DOCUMENT RESUME

ED 344 085 CE 060 908

AUTHOR Craft, Clyde O.
TITLE Computer Graphics. Curriculum Guide for Technology Education.
INSTITUTION Kentucky State Dept. of Education, Frankfort. Office of Vocational Education.
PUB DATE 88
NOTE 196p.; For related documents, see CE 060 907-913.
PUB TYPE Guides - Classroom Use - Teaching Guides (For Teacher) (052)

EDRS PRICE MF01/PC08 Plus Postage.
DESCRIPTORS Competence; Competency Based Education; *Computer Assisted Design; *Computer Graphics; Computer Software; Course Content; Desktop Publishing; Evaluation Criteria; *Laboratory Procedures; *Learning Activities; Lesson Plans; Programing; Secondary Education; Student Evaluation; Teaching Methods; Technical Education; Units of Study
IDENTIFIERS *Apple II; *Basic Programing Language; Technology Education

ABSTRACT

This curriculum guide for a 1-quarter or 1-semester course in computer graphics is designed to be used with Apple II computers. Some of the topics covered include the following: computer graphics terminology and applications, operating Apple computers, graphics programming in BASIC using various programs and commands, computer graphics painting, computer-aided design, computer-aided animation, and desktop publishing. The guide contains a course outline, competencies (task lists), student competency records, and activities. Activities, which serve as lesson outlines used for 1 or more days, include the following: an introduction, an enabling objective, a performance objective, an equipment and supplies list, a performance standard, suggested references for teachers and students, activities, and evaluation criteria. The guide also contains a resource list of 40 books, 9 periodicals, 17 software programs, and 16 videotapes; a glossary; and four appendixes: (1) Apple screen grid sheets, (2) an equipment list, (3) a supply list, and (4) a facility layout. (KC)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED344085

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
☒ This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

Curriculum Guide for Technology Education

COMPUTER GRAPHICS

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

J. Boston

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

Developed by
Clyde O. Craft, Project Director and Writer
Industrial Education and Technology Department
Eastern Kentucky University

for

Kentucky Department of Education
Office for Vocational Education

Curriculum Development Unit
2024 Capital Plaza Tower
Frankfort, Kentucky 40601

1988

CE060908

BEST COPY AVAILABLE

TABLE OF CONTENTS

ACKNOWLEDGEMENTS v

INTRODUCTION vii

Goals viii
Using the Guide ix

COURSE OUTLINE xiii

COMPETENCY LIST xv

STUDENT COMPETENCY RECORD xvii

ACTIVITIES

Computer Graphics Terminology and Applications 1
A Survey of Computer Graphics Applications in Local Industries 15
How to Operate the Apple Computer 19
Graphics Programming in BASIC Using Alphanumeric Characters 25
Graphics Programming in BASIC Using Applesoft Low-Resolution Commands 33
Graphics Programming in BASIC Using Branching and Looping Commands 43
Graphics Programming in BASIC Using Applesoft High-Resolution Commands 57
Graphics Programming in BASIC Using Arrays and Subprograms 69
Graphics Programming in BASIC Using Input Commands 89
Graphics Programming in BASIC Using Output Commands 101
Computer Graphics Painting 109
Computer Aided Design 117
Computer Aided Animation 125
Desktop Publishing 133

LIST OF RESOURCES 141

GLOSSARY 157

APPENDICES

A. Apple Screen Grid Sheets 177
B. Equipment List 185
C. Supply List 189
D. Facility Layout 193

ACKNOWLEDGMENTS

Appreciation is expressed to the following individuals for their contributions to the development of this curriculum guide:

Consultants

Henry H. Lacy, Curriculum Specialist
Curriculum Development Unit
Support Services Division, Office of Vocational Education
Kentucky Department of Education
Frankfort, Kentucky

Jerry Eades, Consultant
Industrial Education Unit
Program Services Division, Office of Vocational Education
Kentucky Department of Education
Frankfort, Kentucky

Steering Committee

Mr. Roger Kincer, Industrial Arts Teacher
Model Laboratory School, Richmond, Kentucky

Mr. Tom Sutton, Industrial Arts Teacher
Western Hills High School, Frankfort, Kentucky

Mr. George Martin, Industrial Arts Teacher
Marion County High School, Lebanon, Kentucky

Mr. Lynn Settles, Industrial Arts Teacher
Jessamine Co Junior High, Nicholasville, Kentucky

Mr. Roy Chapman, Industrial Arts Teacher
Woodford County Middle School, Versailles, Kentucky

Mr. Rick Vincent, Industrial Arts Teacher
Georgetown Middle School, Georgetown, Kentucky

Editors

Dr. Roger Prewitt, Professor
Department of Industrial Education and Technology, Eastern Kentucky University

Mr. Jim Brown, Office of Vocational Education

Special thanks to the following individuals who field tested selected activities from this guide in their classes:

Mrs. Betty Collins, Industrial Arts Teacher
Breathitt County High School, Jackson, Kentucky.

Mrs. Doris Lash, Math and Computer Science Teacher
Madison Central High School, Richmond, Kentucky

Mr. Lynn Settles, Industrial Arts Teacher
Jessamine Co Junior High, Nicholasville, Kentucky

Computer Graphics for Technology Education

Introduction

This curriculum guide was developed for technology education students and teachers interested in exploring computer graphics; a very important, exciting, and special segment of *communications technology*. Computer graphics is a collection of technologies that use the computer to generate and communicate information coded as graphics--drawings, captured video images or photographs, stylized text, or combinations of these. Computer graphics applications include the utilization of the computer for desktop publishing, design and drafting, animation, business graphics, medical graphics, cartography, and art. Other applications are being added yearly, if not more frequently.

Students of communications know that all communications systems include a source, the transmitted information (code), and the destination. The study of computer graphics is an excellent way to explain some of these conceptional parts of a communications system (Seymour, Ritz, Cloghessy, 1987, p.6).

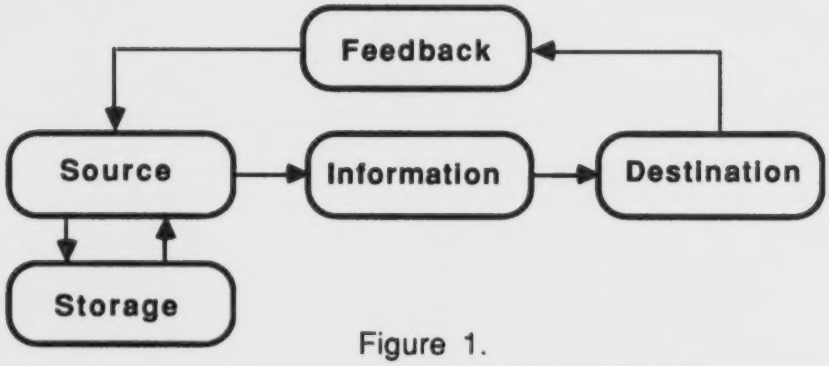


Figure 1.
Communications Model

The computer is a device that can be used in communications as the *source of information*. Examples include computers that are used in flight simulators. They create new views of the landscape and display them in the windows of the simulated airplane cockpit. This example also illustrates very well the use of *feedback* in a communications system. The views seen by the simulator pilot are created instantaneously as he or she reacts to a given situation. As the flight controls are adjusted by the pilot, the new scenes appear in the cockpit window just as they would in a real aircraft under similar conditions.

In computer graphics, the computer is also used to transmit information. The computer has characteristics that allow it to store and transmit graphic information in unique ways. Images can be stored and transmitted at a very high speed. Sometimes as many as 30 images per second can be displayed to the user--so fast, that the human eye cannot detect that separate images are being displayed. The result is real time movement or animation similar to the movies. In computer aided drawing (CAD) applications, the speed of the computer to react to human feedback allows the user

to edit drawings much quicker than has ever been possible before. The benefit of using such an application is increased productivity for manufacturing and construction industries.

Computer graphics can be an exciting field for technology students to explore. And, because computer graphics promises to play such a vital role in the *information age*, this exploration can also help students become more technologically literate.

Goals

The goals of this curriculum guide are to develop:

a) an interest in computing, especially in computer graphics;

b) an in-depth understanding of the concepts of communication technology through the study of one special communication field--computer graphics;

c) skills and knowledge in computer graphics programming and the use of computer graphics applications;

d) self-realization and initiative by providing a unique outlet for students' talents;

e) creative abilities through the medium of computer graphics; and

f) an appreciation for the talents of computer graphics professionals and for the products of their work.

These goals are offered for guidance to the technology teacher when using this curriculum guide. They are much more general than the objectives provided with each activity which are most nearly related to goal c. The activity objectives are limited to those behaviors that are easily measured, generally cognitive and psychomotor behaviors. The goals, on the other hand, are written in much more subjective terms. Therefore, they will be interpreted differently by each teacher using them. Although determining how well you have achieved them is less precise, they are no less important. In fact, for the long range benefit of the student goals a,b,d,e, and f are much more important than goal c and the related activity objectives.

For example, when selecting activities for an individual student, the most important considerations are the student's interest (goal a) and likelihood of his or her success (goal d). Select an activity that is most likely to capture his or her interest and one that he or she will be most likely to be successful with. After interest has been developed and the student has been successful, it will be easier for them to proceed to less interesting or more difficult activities. However, the student's interest should never be sacrificed just because an activity has not been completed. It is better to develop an alternate and more interesting activity for the student than to force him or her to use one from this guide that is of no interest to him or her. In summary, it would be better not to teach this course at all than to teach it in a manner where activity objectives were achieved but in the process, most of the students lost all interest in the subject.

In many cases, the achievement of these goals will be more dependent on the approach taken by the teacher in his or her teaching than any other factor. If the teacher is enthusiastic and excited about computer graphics, then it will carry over to the students and they will become excited about

it. In turn, they will have more interest, progress more easily through the activities (goal d), be more creative (goal e), and have a greater appreciation for the talents and products of computer graphics professionals (goal f).

Using the Guide

This curriculum guide was written to be used with the Apple II family of computers because of the popularity of these machines in the secondary schools of Kentucky. Those teachers using other types of computers can use the guide; however, they will need need to make extensive modifications in the programming exercises found in activities COG-003 through COG-010. It will also be necessary for them to select different references for these activities. For activities COG 011 through COG-014, different brands of application software will need to be selected. Even so, the guide can give the non-Apple using teacher much direction.

A 180 day offering. When using the activity guides for a 180 day offering, the activities can be presented to the students in several ways. First of all, teachers are encouraged to develop their own activities for the course. However, if none have been developed, then all of the activities in the guide will need to be used. Activities COG-001 & COG-002 should be spread over the entire 180 day period. Students should be directed to work on them a little at a time with a lot of help from the teacher. Activity COG-003 should be completed before working on any of the remaining activities. Activities COG-004 through COG-010 involve programming and should be completed in sequence. The remaining activities, COG-011 through COG-014, can be completed in any order but it is more logical for COG-011to proceed COG-013 or COG-014. There is no reason for not mixing the application activities and the programming activities, that is, a student who is working in the programming activities COG-004 through COG-010 could skip to one of the application activities before proceeding to complete the programming sequence.

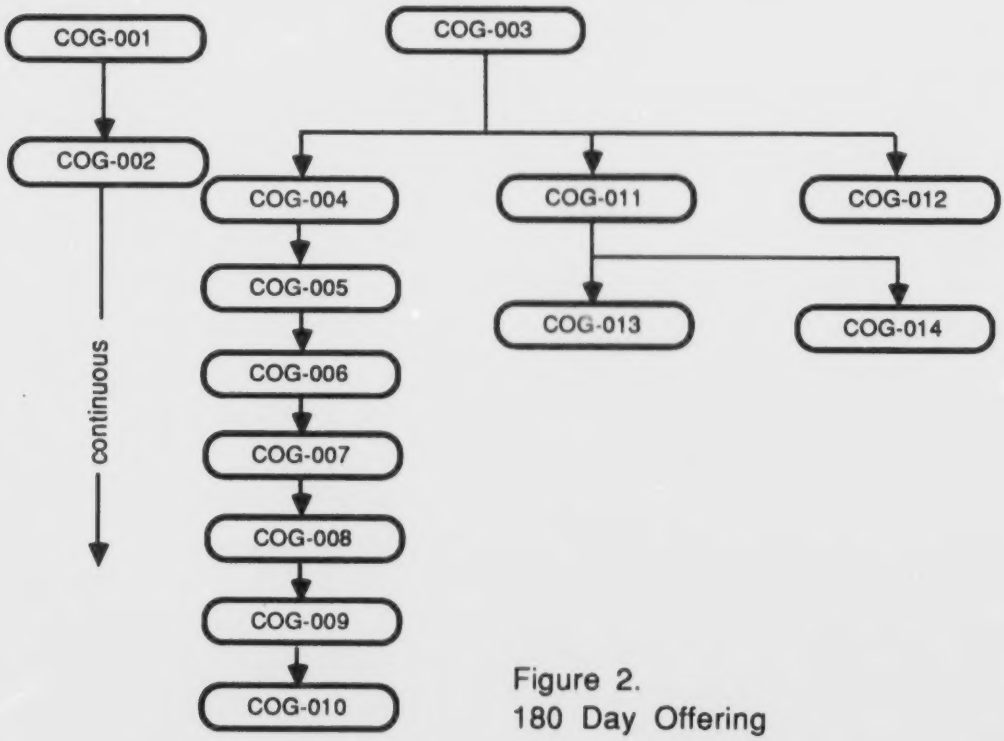


Figure 2.
180 Day Offering

A 90 day offering. There are several approaches if the teacher wishes to use the guide in a 90 day offering. Following the sequences explained above, the teacher could offer a 90 day course involving only the programming activities--COG-004 through COG-010. Another option would be to offer only the application activities. A third option would be to mix the two. For example, the teacher may have his or her students work through half of the programming activities and two of the application activities. With all of these options, a portion of COG-001 and COG-002 and all of COG-003 should be included.

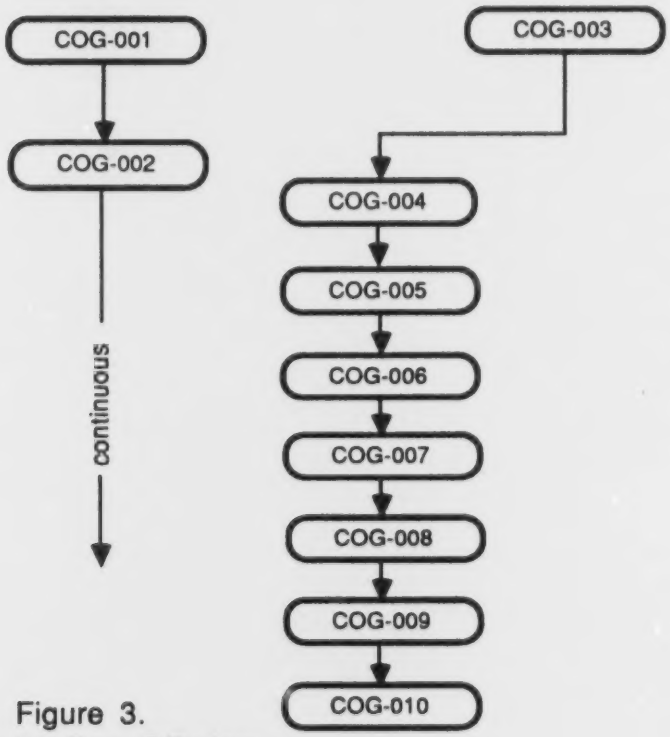


Figure 3.
90 Day Offering

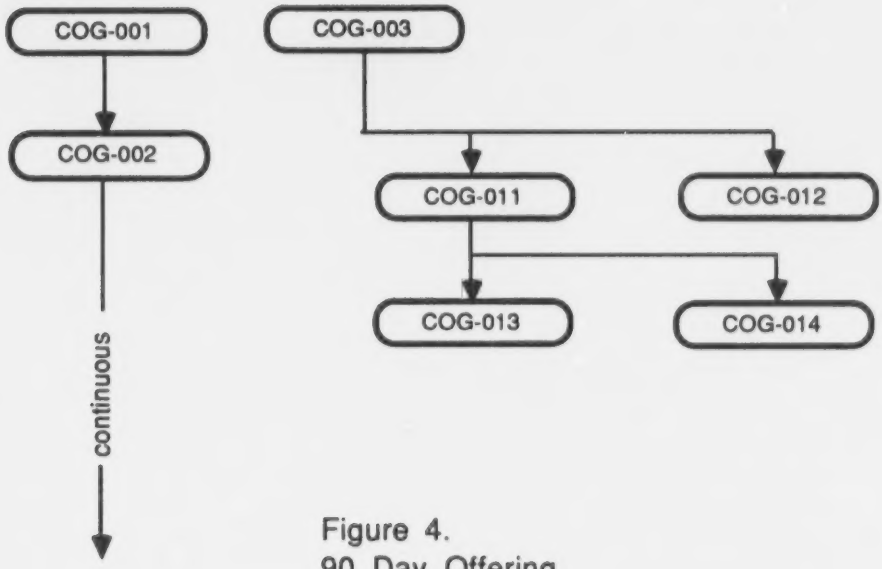


Figure 4.
90 Day Offering

Using the guide within other technology education classes. Yet another way the guide can be used is to select certain activities from it for use in other classes. For example, a teacher offering a traditional class in graphic arts or a technology education class in communications may elect to use the desktop publishing activity from this guide in his or her class or a teacher offering a survey of engineering may elect to use the CAD activity from this guide in his or her class. If this is done, the COG-003 activity and portions of the COG-001 activity may also be needed. Teachers can also use the activities from this guide for individual students enrolled in "special problems" types of courses or individual students in other courses needing advanced or more challenging activities. Students given the activities to work through on their own, should be good readers and self starters.

References

Seymour, R.D., Ritz, J. M., & Cloghessy, F.A. (1987). *Exploring communications*. South Holland, IL: Goodheart-Willcox Co.

Course Outline

I. Introduction to the Course.
- A. How the course will be conducted.
- B. Care and use of the computers.
- C. How students will be evaluated.

II. Computer Graphics Overview.
- A. History of computer graphics.
  1. famous personalities in computer graphics.
  2. hardware time-line and its effect on computer graphics.
  3. software time-line and its effect on computer graphics.
  4. the future and computer graphics.
- B. Terminology
  1. general computer.
  2. computer graphics.
- C. Overview of applications
  1. (CAD) computer aided design.
  2. computer paint systems.
  3. desktop publishing.
  4. business graphics
  5. computer graphics in scientific research.
  6. computer graphics in medicine.
  7. computer animation in movies and T.V.
  8. computer and video games.

III. Computer Graphics Programming on the Apple.
- A. Graphics programming with alphanumeric characters.
- B. Low resolution graphics programming.
- C. High resolution graphics programming.
- D. Programming graphics input devices.
- E. Programming graphics output devices.

IV. Using Computer Graphics Applications.
- A. (CAD) computer aided drafting and design.
- B. Computer paint systems.
- D. Desktop publishing with a computer.
- E. Computer animation.

Computer Graphics Competencies

Duty: **Computer Graphics General** 15 days spread over 90 or 180 days

The student will be able to:

COG-001 Define computer graphics terminology, trace the development of computer graphics, and describe computer graphics applications.

COG-002 Describe computer graphics applications found in local businesses and industries.

Duty: **Basic Procedures for Operating a Microcomputer.** 2 days

The student will be able to:

COG-003 Perform basic operations on an Apple computer.

Duty: **Computer Graphics Programming on an Apple II** 73 days

The student will be able to:

COG-004 Write a computer graphics program using Applesoft variable names and the Applesoft BASIC commands HTAB, VTAB, HOME, LET, PRINT, REM, and END.

COG-005 Write a low resolution graphics program using the Applesoft BASIC commands GOTO, GR, PLOT, HLIN, VLIN, COLOR, INPUT, READ, and DATA.

COG-006 Write a low resolution graphics program using the Applesoft BASIC commands GET, IF...THEN, FOR...TO and NEXT, and ON...GOTO.

COG-007 Write a high resolution graphics program using the Applesoft BASIC commands HGR, HCOLOR, HPLOT, PEEK, POKE, and CALL.

COG-008 Write a high-resolution graphics program using arrays, functions, and the BASIC commands GOSUB...RETURN, DIM and DEF FN.

COG-009 Write an interactive graphics program which utilizes graphics input devices.

COG-010 Write a program which outputs a graphic image to a graphic output device (pen plotter).

Duty: **Microcomputer Graphics Applications** 73 days

The student will be able to:

COG-011 Describe a computer paint system application and use one to paint computer generated images.

COG-012 Describe a 2-D computer aided drafting system application and use one to produce technical drawings.

COG-013 Describe computer generated animation applications and produce an animation sequence.

COG-014 Describe a publishing system application and use one to produce camera copy for publication.

Total for all activities 180 days

STUDENT COMPETENCY RECORD

School _____ Student Address _____

Student _____ _____ _____ Zip _____

School Year _____ Comments _____

Instructor's Name _____

Course Title _____

Classification _____

The student should be evaluated according to the performance ratings listed. Indicate on this form the numeric rating of the student's performance for each item.

(1) Did not attempt the activity.
(2) Could not perform the tasks according to the evaluation criteria.
(3) Performed the tasks according to the evaluation criteria.

COMPETENCIES

	COG-001	Define terminology, trace the development of and describe applications related to computer graphics.
	COG-002	Describe computer graphics applications found in local businesses and industry.
	COG-003	Perform basic operations on an Apple computer.
	COG-004	Write a computer graphics program using variables & HTAB,VTAB,HOME,LET,PRINT,REM &END
	COG-005	Write low-res program using GOTO,GR,PLOT,HLIN, VLIN,COLOR,INPUT,READ, & DATA.
	COG-006	Write low-res program using GET, IF..THEN, FOR...TO, NEXT, & ON...GOTO.
	COG-007	Write hi-res program using HCOLOR,HPLOT,PEEK, POKE, & CALL.
	COG-008	Write a hi-res program using arrays, functions, and GOSUB...RETURN,DIM, & DEF FN.
	COG-009	Write an interactive program that utilizes graphic input devices.
	COG-010	Write a program that outputs a graphic image to an output device.
	COG-011	Describe a computer paint system application and use one to paint a computer generated image.
	COG-012	Describe a 2D computer aided drafting system application and use one to produce a drawing.
	COG-013	Describe computer generated animation applications and produce an animation sequence.

	COG-014	Describe a publishing system application and use one to produce camera copy for publication.

Computer Graphics General

COMPUTER GRAPHICS TERMINOLOGY AND APPLICATIONS

Introduction: This activity is designed to introduce the student to computer graphics, the terminology associated with this technology, and the many applications for which computer graphics is presently being used. It should be the first activity given to the student and the last for which the student is evaluated. It is an activity designed to be worked on throughout the entire course of instruction.

Performance Objective (10 days spread over 90 or 180 days):

TASK/COMPETENCY--COG-001

Upon completion of this activity, the student will be able to define computer graphics terminology, trace the development of computer graphics, and describe computer graphics applications. More specifically, the student should be able to define the following terms, acronyms, and abbreviations:

general computer terminology:

alphanumeric
application software
assembly language
binary
bit
byte
CPU
CRT
FORTRAN
hardware
hexadecimal
high level language
I/O device
kilobyte
mainframe
mass storage device
megabyte
menu
microcomputer
microprocessor
minicomputer
PASCAL
PROM
software
workstation

terminology unique to computer graphics:

aliasing
antialiasing
B-spline
Cartesian coordinate system
computer animation
computer graphics
computer graphics system
diffused light source
digitizer
display device *
dot matrix printer
drum plotter
electronic air brush
electrostatic plotter
faceted polygons
flatbed pen plotter
graphics tablet
image enhancement
joystick
light pen
modeling
mouse
National Computer Graphics Association
passive graphics
pattern recognition
puck
raster refresh
rendering
shaded images
simulation
solids model
storage tube
thumbwheels
track ball
transformation
wire frame model
X axis
Y axis

these terms will be covered in COG-003:

display device *
file
floppy diskette
format a disk
hard disk
LOAD
matrix printer
operating system

program
RUN
SAVE
volume

these terms will be covered in COG-004:

BASIC
compiler
interpreter
machine language
object code
RAM
ROM
source code

these terms will be covered in COG-005:

Cartesian coordinates
pixel *
resolution

these terms will be covered in COG-006:

aspect ratio
batch mode
fatal error
interactive

these terms will be covered in COG-007:

bit map
display buffer
frame buffer * (framestore in England)
high resolution graphics display buffer

these terms will be covered in COG-008:

array
clipping
end point
graphic transformation
matrix
point of intersection
window

these terms will be covered in COG-009:

input constraints

these terms will be covered in COG-010:

viewport
absolute coordinates

relative coordinates
"A", "B", and "C" size paper

these terms will be covered in COG-011:

palette
shade
drag
shadow
cut and paste
pixel *
fill pattern
fat bits
fill pattern leak
basic colors of light

these terms will be covered in COG-012:

absolute coordinate *
AEC
CAD
CAD function
Cartesian coordinates *
CIM
clipping *
cursor
digitizing
grid
IGES
input
interactive *
isometric drawings
jaggies
layers
mirror
origin
orthographic drawings
output
perspective drawings
polyline
relative coordinate *
resolution *
viewport *
wireframe
window *
zoom

these terms will be covered in COG-013:

flicker
flip book
frame buffer * (framestore in England)
object blend sequence
persistence of vision

pixel *
real time animation
storyboard
thaumatrope

these terms will be covered in COG-014:

binding
greeking
justification
kerning
leading
letterspacing
offset printing
optical character readers
photocopying
proofreading
text fonts
scanners
snap-to
WYSIWYG

* *Since students may not be required to complete all activities, some terms are listed twice.*

CONDITIONS

The above tasks should be performed under the following conditions:

a) The student should be provided twenty terms selected at random. The student will be asked to provide definitions for these terms from memory under test conditions.

b) The student should be expected to trace the development of computer graphics, and describe computer graphic applications with the help of printed references.

STANDARDS OF PERFORMANCE

The above tasks should be performed to the following standards:

a) The student should be expected to provide correct definitions for 70% of the terms (a term will be considered correctly defined if the definition provided is similar to one found in the current computer literature).

b) The student should be expected to describe at least three milestones (describe the event, the approximate date, and the personalities, hardware, or software involved) in the evolution of computer graphics; and describe two applications of computer graphics to include the benefits of the technology, the training needed to use the application, and the type of hardware used in the application.

Selected Instructional Materials: A videotape recorder and the videotapes and books listed below.

Teacher References

Recommended

Books:

Hubbard, S. W. (1983). *The computer graphics glossary*. Phoenix, AZ: Oryx Press.

Jankel, A., & Morton, R. (1984). *Creative computer graphics*. Cambridge, England: Cambridge University Press.

Spencer, D. D. (1986). *The illustrated computer dictionary (3rd ed.)*. Columbus, OH: Merrill.

Videotapes:

Electronic Pictures Corporation. (1983). *Creative computer graphics*. Available from Frost and Sullivan, Inc., Department R-1, 106 Fulton Street, New York, NY 10038.

Herr, L., & Zaritsky, R. (1988). *Visualization: state of the art. A special issue of the ACM/SIGGRAPH video review--volume 30*. Association for Computing Machinery, 11 West 42nd Street, New York, NY 10031.

Suggested

Books:

Consumer Guide Editors. (1983). *Illustrated computer dictionary*. Skokie, IL: Publications International.

Winkler, C. (1983). *The computer careers handbook*. New York: Arco Publishing Co.

Teacher Activities

1. View the videotape *Creative Computer Graphics* and familiarize yourself with the book *Creative Computer Graphics* by Jankel and Morton.

2. Introduce the videotape and the book to your students. Go over the questions found in the student activities with your students before they view the videotape for the first time. However, advise them that they can view it again if they don't remember the answers to the questions after the first viewing. You may need to help them find the answers during the second viewing by stopping and starting the tape at the appropriate places.

Explain to them that the book was published in England and that some words are spelled differently than in the U.S. or terms may be different. Examples: colour/color, pounds/dollars, computer cash point/automatic bank teller, etc.

There are computer generated images of nudes on pages 18 and 84. They are rather abstract and should not offend anyone. However, you may want to check with your principal and librarian before making any assignments from the book.

Although this is the logical activity with which to begin the course, it is not intended that students finish it before going on to other activities. Students should be allowed to work on this activity throughout the course. You may choose to introduce them to computer graphics with the videotape, have them answer those questions that they can after the first viewing, and then allow them to return to the activity for short periods throughout the course. *Do not force them to stay with it until they become bored.*

3. Become familiar with all of the terms found in Performance Objective. Introduce them to the students a few at a time throughout the course. This can be done in a variety of ways: have a short discussion of a different term at the beginning of each class period; each week, have the students find the definition of a different term on their own; or use the terms to devise a crossword puzzle for them to work with.

4. Preview Section 1, "Interactivity," in the videotape *Visualization: State of the Art* and study Chapter One," History," in Jankel and Morton's *Creative Computer Graphics*. Work very closely with your students as they trace the history of computer graphics. You may want to lead a discussion, make a presentation, or have students make a report on the history of computer graphics.

5. Have individual students or small groups of students make reports throughout the year on different computer graphics applications--computer aided design, animation, desktop publishing, business graphics, etc. Material for these reports can come from *Creative Computer Graphics*, computer graphics periodicals such as *Computer Graphics World*, or from visits to local businesses and industries marketing or using the application (see Activity COG-002). Preferably, these reports would precede the activity where that application is experienced.

Student References

Recommended

Books:

Hubbard, S. W. (1983). *The computer graphics glossary*. Phoenix, AZ: Oryx Press.

Jankel, A., & Morton, R. (1984). *Creative computer graphics*. Cambridge, England: Cambridge University Press.

Spencer, D. D. (1986). *The illustrated computer dictionary (3rd ed.)*. Columbus, OH: Merrill.

Videotapes:

Electronic Pictures Corporation. (1983). *Creative computer graphics*. Available from Frost and Sullivan, Inc., Department R-1, 106 Fulton Street, New York, NY 10038.

Herr, L., & Zaritsky, R. (1988). *Visualization: state of the art. A special issue of the ACM/SIGGRAPH video review--volume 30*. Association for Computing Machinery, 11 West 42nd Street, New York, NY 10031.

Suggested

Book:

Consumer Guide Editors. (1983). *Illustrated computer dictionary*. Skokie, IL: Publications International.

Winkler, C. (1983). *The computer careers handbook*. New York: Arco Publishing Co.

Student Activities

1. Following an introduction by your teacher, view the videotape *Creative Computer Graphics*. After viewing the tape, answer the following questions. If necessary, view the tape a second or third time.

   a. What are some simulation applications for computer graphics?

   _____

   _____

   _____

   b. How can computer graphics help architects, designers, and planners?

   _____

   _____

   _____

   c. Give some examples of computer graphics applications in the field of science.

   _____

   _____

   _____

d. What are the advantages of CAD (computer aided design) over traditional designing techniques?

e. What two techniques of modeling (object description) are discussed in the videotape?

f. What are the two types of models?

g. What is a pixel?

h. What are the three types of transformations discussed on the videotape?

i. If you transformed an image by translating it along the X axis, describe how it would change.

k. Is the Z axis necessary when using a 2D system?

1. What is a computer graphics system? Name the four types discussed on the videotape.

_____

_____

_____

m. Where is a storyboard used?

_____

_____

_____

2. Read Chapter One of *Creative Computer Graphics* by Jankel and Morton. Then answer the following questions:

a. How do the authors define computer graphics? They say it is comprised of . . .

_____

_____

_____

b. The authors say "Every use of a computer involves, to a greater or lesser extent, engaging in a dialogue with it." Give at least two examples of how a person engages in dialogue with a computer.

_____

_____

_____

c. What is simulation? Why is computer graphics such a great partner for simulation?

_____

_____

_____

d. Tell of some of the work of Dr. James F. Blinn.

_____

_____

_____

e. What was the first motion picture to incorporate a large proportion of computer generated images?

f. Explain what the authors mean when they say, ". . . the essence of computer graphics is communication."

3. Read Chapter Two of *Creative Computer Graphics* by Jankel and Morton. Then answer the following questions:

a. What was the first production system that relied on the use of interactive computer graphics?

b. Who was the father of real time interactive computer graphics? Tell of his accomplishments.

c. Describe the impact of *Sketchpad.*

d. What is meant by the terms vector display, refresh, and flicker?

e. What one factor more than any other has accounted for the increased use of raster refresh display devices for computer graphics.

_____

_____

_____

f. Tell of the contributions of John Whitney Sr.

_____

_____

_____

g. What did Frank Sindon's film *Force, Mass, and Motion* demonstrate?

_____

_____

_____

h. What corporation was, in large part, responsible for introducing computer graphics to T.V.? Tell about this introduction.

_____

_____

_____

i. The images used by T.V. weatherpersons is a product of what form of computer graphics?

_____

_____

_____

j. What were the contributions made by *Pong* and *Space Invaders* ?

_____

_____

_____

4. View "Section 1: Interactivity" of the videotape, *Visualization: State of the Art*. In this videotape section there is a historical segment in which Stephen Coons is interviewed. Made in 1964, this interview documents the early work of Ivan Sutherland and others when they were working on the project *Sketchpad* at M.I.T's Lincoln Laboratory. *Project Sketchpad* is considered by most to be the beginning of interactive computer graphics. Therefore, this video segment has a great deal of historical significance.

5. Learn the definitions of the following terms. You can learn them through discussions with your teacher or through your own individual readings. Use the references listed above.

general computer terminology:

alphanumeric
application software
assembly language
binary
bit
byte
CPU
CRT
FORTRAN
hardware
hexadecimal
high level language
I/O device
kilobyte
mainframe
mass storage device
megabyte
menu
microcomputer
microprocessor
minicomputer
PASCAL
PROM
software
workstation

terminology unique to computer graphics:

aliasing
antialiasing
B-spline
Cartesian coordinate system
computer animation
computer graphics
computer graphics system
diffused light source
digitizer
display device *
dot matrix printer
drum plotter
electronic air brush
electrostatic plotter
faceted polygons
flatbed pen plotter
graphics tablet
image enhancement
joystick
light pen
modeling

mouse
National Computer Graphics Association
passive graphics
pattern recognition
puck
raster refresh
rendering
shaded images
simulation
solids model
storage tube
thumbwheels
track ball
transformation
wire frame model
X axis
Y axis

Evaluation Criteria: To evaluate the student's success in completing this activity, the teacher should design and administer a written cognitive test. The test should consist of three parts (the teacher could also decide to administer three separate tests). One part should require the student to provide the definitions for twenty of the terms covered in this activity selected at random. A second part of the test should require the student to trace the evolution of computer graphics by describing major milestones (describe the event, the approximate date, and the personalities, hardware, or software involved). A third part should require the student to describe several applications of computer graphics. These descriptions should include the benefits of the technology, the training needed to use the application, and the type of hardware used in the application.

To be considered successful, the student should be able to provide acceptable definitions for 14 of the 20 terms, provide descriptions for at least three milestones in the evolution of computer graphics, and be able to provide descriptions of two computer graphics applications.

Computer Graphics General

A SURVEY OF COMPUTER GRAPHICS APPLICATIONS IN LOCAL INDUSTRIES

Introduction: This activity is designed to introduce the student to computer graphics applications that can be found in the local community or the local region of the state. The student can be assigned this activity anytime after he or she has been introduced to computer graphics and the many computer graphic applications as described in Activity COG-001, Computer Graphics Terminology and Applications.

Enabling Objective: The student should have completed that portion of Activity COG-001 that deals with an overview of computer graphics applications.

Performance Objective (4-5 days spread over 90 or 180 days):

TASK/COMPETENCY--COG-002

The student should be able to describe computer graphics applications found in local businesses and industries.

CONDITIONS

Under test conditions, the student should be able to recall from memory, at least one computer graphics application found in the local community or region of the state.

STANDARDS OF PERFORMANCE

The student's description should: a) classify the application into one of the major application areas discussed in Activity One, b) identify the type of computer equipment being used (by brand name and type--micro, workstation, mini, mainframe), c) describe the benefits that the company has accrued from its use, d) describe the source of the software, and e) describe the educational and career background of the operators who are using the system.

Teacher References

Recommended

Books: Jankel, A., & Morton, R. (1984). *Creative computer graphics*. Cambridge, England: Cambridge University Press.

Suggested

Book: Winkler, C. (1983). *The computer careers handbook*. New York: Arco Publishing Co.

Teacher Activities

1. Identify local or regional businesses or industries that are using computer graphics applications. Below are some ways that may help you identify them:

   a. Visit local industries in an effort to identify any that are using computer graphics applications. Do not forget T.V. stations, newspaper offices, independent printing houses, and engineering or architectural offices, as well as, manufacturing and construction concerns.

   b. Contact local computer clubs and organizations, especially the local chapter of the National Computer Graphics Association, for information about users.

   c. Check with the local Chamber of Commerce or local computer stores to see if they can guide you to some businesses that use computer graphics.

2. Once you have identified those businesses and industries using computer graphics, make arrangements for a class field trip, a field trip for a small group of your students, or for someone from the business or industry to visit your school to make a presentation about their application.

3. If permitted by the company you are visiting, take along a video camera and record your visit. This could be useful for reviewing the trip with your class or for subsequent classes.

Student References

Recommended

Books: Jankel, A., & Morton, R. (1984). *Creative computer graphics*. Cambridge, England: Cambridge University Press.

Suggested

Book: Winkler, C. (1983). *The computer careers handbook*. New York: Arco Publishing Co.

Student Activities

1. After your teacher announces that you are going to take a field trip or have a guest speaker from a local business or industry, go back and review the particular application (CAD, business graphics, animation) used by this company. The best source may be Jankel and Morton's *Creative Computer Graphics*.

2. Work up a set of questions for the event. These questions will represent what you would like to find out about the company's particular application during the visitation or presentation. Your questions should include questions about the type of hardware they use, where they acquired the software, how they trained their operators, their operator's educational background, and to what extent the company finds the use of computer graphics to be beneficial.

3. After the presentation or when you return from the field trip, write a newspaper article about the trip. You can then submit it to the school newspaper, the technology club's newsletter, the community paper, or just give it to your teacher so that she or he may award you extra credit.

4. Write a letter to the person who made the presentation or who acted as the host for your field trip thanking them for their efforts.

5. If you have a close friend of the family or a relative whose occupation involves working with computer graphics, ask them if it would be possible for you to go to work with them and follow them around for a day or a half day. If so, make arrangements with your teacher and principal to be away from school. Be sure to decide beforehand what you want to learn, so that when you return to school, you can share your experience with your classmates and teacher.

Evaluation Criteria: To evaluate the students' success in completing this activity, the teacher should give them a short essay quiz on which the students should be expected to describe the company/s they visited or the presentation/s they heard.

To be considered successful on the essay quiz, the student should be able to: a) classify the application into one of the major application areas discussed in Activity One, b) identify the type of computer equipment being used (by brand name and type--micro, workstation, mini, mainframe), c) describe the benefits that the company has accrued from its use, d) describe the source of the software, and e) describe the educational and career background of the operators who were using the system.

Basic Procedures for Operating a Microcomputer

HOW TO OPERATE THE APPLE COMPUTER

Introduction: This activity is designed to introduce the student to the Apple computer and to its proper operation. This activity can be done on any Apple computer that uses Applesoft BASIC, the II+, IIe, IIc, or IIgs.

Performance Objective (2-3 days):

TASK/COMPETENCY--COG-003

The student should be able to perform basic operations on an Apple computer. More specifically, he or she should be able to:

a) turn the computer, monitor, and printer on and off;

b) format a blank diskette and make a backup copy of a diskette on which files have been saved;

c) LOAD, SAVE, DELETE, CATALOG, and RUN programs stored on a diskette using either Apple DOS or ProDOS;

d) direct output to both the printer and the display device to get a listing of a program;

e) use the keyboard to input any alphanumeric string;

f) use the special function keys; and

g) define the following new terms associated with the operation of an Apple computer:

operating system	format a disk
volume	file
floppy diskette	program
hard disk	matrix printer
SAVE	display device
LOAD	RUN

CONDITIONS

The above tasks should be performed under the following conditions:

a-f) The student should be provided an Apple computer with a disk drive, monitor, printer, a blank diskette, and a master diskette on which programs have been stored.

g) Terms should be recalled from memory.

STANDARDS OF PERFORMANCE

The above tasks should be performed to the following standards:

a-f) To be considered acceptable, all the above operations must be done as described in the proper Apple reference manual and have the intended results.

g) Correct definitions (as defined in an Apple reference manual) of at least 4 of 5 randomly selected terms should be considered acceptable.

Selected Instructional Materials: An Apple II+, IIe, IIc, or IIgs computer with printer, monitor (preferably with color), and disk drive; an Apple II utilities diskette or system master diskette; a blank diskette; the owner's manual for the Apple computer being used, the *Computer Graphics Curriculum Project Master Diskette*, and one of the following: *Your Tour of the Apple IIgs, Apple Presents...Apple,* or *Apple Presents the IIe/An Introduction.*

Teacher References

Recommended

Books:

When using the Apple DOS operating system:

Apple Computer, Inc. (1983). *Apple II DOS user's manual.* Cupertino, CA: Author.

or

Apple Computer, Inc. (1983). *Apple IIe owner's manual.* Cupertino, CA: Author.

or

Any other appropriate Apple II owner's manual.

When using the Apple ProDOS operating system:

Apple Computer, Inc. (1983). *Apple II ProDOS user's manual.* Cupertino, CA: Author.

or

Apple Computer, Inc. (1987). *BASIC programming with ProDOS.* Reading, MA: Addison-Wesley.

Software:

Apple IIgs System Disk, ProDOS User's Disk, or Apple II System Master (DOS 3.3).

Videotapes: DCA Educational Products, Inc. (1982). *Apple IIe, VHS*. from the *Computer Operations Series*. Bedminster, Pennsylvania 18910: Author.

Suggested

Book: *Apple II Microcomputer Quick Reference Guide*.

Teacher Activities

1. Using the proper manual, familiarize yourself with the proper procedure for: a) turning the computer, monitor, and printer on and off; b) formatting and copying a diskette; c) cataloging, loading, saving, and running a program to or from a diskette; d) directing output (listing) to a printer and to the display screen; and e) special features of the keyboard to include special function keys.

2. Demonstrate the above topics to your students, have them view the videotape *Apple IIe, VHS*, or have them work through a self-instructional module designed to teach them the above skills.

3. Your demonstration or self-instructional module should also include a discussion or treatment of the new computer terminology being used in this activity (see objective #2 above).

4. If your computer came with a CAI program that introduces the computer and its keyboard, have the students work through all parts of it--*Your Apple Tour of the Apple IIgs, Apple Presents...Apple*, (for the Apple IIe), *Apple Presents the IIe/An Introduction* (for the enhanced Apple IIe), etc. Note: the programs *Apple Presents...Apple* and *Apple Presents the IIe/An Introduction* are similar; however, the latter program teaches the use of the DELETE key as it should be used with an application program, such as *AppleWorks*, not as it is used when writing BASIC programs, where it cannot be used. Be sure to caution your students about this.

5. Supply each student with (or have them buy) a 5 1/4" (3 1/2" can be used if the school has an Apple IIgs) diskette.

6. Supply your students with the programs, "Computer Graphics Demo I and II", which can be found on the *Computer Graphics Curriculum Project Master Diskette* (COG.Project/...).

Student References

Recommended

Books: *When using the Apple DOS operating system:*

Apple Computer, Inc. (1983). *Apple II DOS user's manual*. Cupertino, CA: Author.

or

Apple Computer, Inc. (1983). *Apple IIe owner's manual*. Cupertino, CA: Author.

or

Any other appropriate Apple II owner's manual.

When using the Apple ProDOS operating system:

Apple Computer, Inc. (1983). *Apple II ProDOS user's manual*. Cupertino, CA: Author.

or

Apple Computer, Inc. (1987). *BASIC programming with ProDOS*. Reading, MA: Addison-Wesley.

Software: *Your Tour of the Apple IIgs, Apple Presents...Apple*, or *Apple Presents the IIe/ An Introduction.*

The programs "Computer Graphics Demo I and II" from the *Computer Graphics Curriculum Project Master Diskette.*

Apple IIgs System Disk, ProDOS User's Disk, or *Apple II System Master (DOS 3.3).*

Videotapes: DCA Educational Products, Inc. (1982). *Apple IIe, VHS*. from the *Computer Operations Series*. Bedminster, Pennsylvania 18910: Author.

Suggested

Book: *Apple II Microcomputer Quick Reference Guide.*

Student Activities

1. Using the proper reference from those listed above, read about the following:
   a. How to turn the computer, monitor, and printer on and off.
   b. How to format and copy a diskette.
   c. How to LOAD, SAVE, and RUN a program that is stored on a diskette.

d. How to get a listing of the programs stored on a diskette, using the CATALOG command or the Apple ProDOS System Utilities.

e. How to use PR#1 to direct output (a listing) to a printer and PR#0 to return output back to the display screen.

2. See your instructor to receive a demonstration or a self instructional module or view the videotape *Apple IIe, VHS* covering the general operation of an Apple computer and all the topics you read about in 1 above.

4. Get the diskette *Your Apple Tour of the Apple IIgs* if you are using an Apple IIgs, *Apple Presents...Apple* if you are using an Apple IIe, or *Apple Presents the IIe/An Introduction* if you are using an enhanced IIe. Place the diskette in Drive 1 and turn the machine on. Work through the entire program and HAVE FUN!!!

5. Get a blank diskette from your instructor. Format the diskette. If you are using the ProDOS operating system, use the formatting utility on the *Apple IIgs System Disk* or the *ProDOS User's Disk*. If you are using the DOS 3.3 operating system, key the following program into the computer's memory and use it to format your diskette using the INIT command. When keyboarding the program into the computer, if you make a mistake and do not remember how to correct it, simply key in the line again. If you have a line you wish to erase completely, key in the line number and then press the carriage return key.

[COG.003.1]

```
NEW

10 REM  ** HELLO PROGRAM **
20 PRINT "COMPUTER GRAPHICS"
30 PRINT "PROGRAMS"
40 PRINT: PRINT
50 PRINT "student's name"
60 PRINT "school's name"
70 PRINT "date"
80 END
```

Be sure that you have the blank diskette you wish to format in the disk drive. If you are unsure, check with your instructor.)

```
INIT HELLO
```

6. Get a copy of the *Computer Graphics Curriculum Project Master Diskette* from your instructor. One at a time, load the programs, "Computer Graphics Demo I" and "Computer Graphics Demo II." Run each of the programs at least once. Get a listing of one of the programs on your printer. Save both of the programs to your diskette. Send a listing of the programs on your diskette to the printer using the CATALOG command (it will work with either DOS or ProDOS).

7. Using your other blank diskette, make a backup copy of the diskette which you formatted and on which you saved the two demonstration programs (see 6 above). In making your backup copies, use COPYA (for DOS 3.3) on the *Apple II System Master* diskette or the copy utility on *Apple ProDOS System Utilities* diskette.

Evaluation Criteria: To evaluate the student's success in completing this activity, the teacher should observe the student while she/he does those tasks listed below. Successful completion of the tasks should be measured by how well the student follows the procedure described in the proper Apple manual and whether the intended is accomplished.

1. Have the student turn the computer and monitor on, load a program and run it.

2. Have the student list the program on the printer. Then, have the student turn the printer off and get a listing of the program on the display screen.

3. Have the student format a blank diskette, save a program to the diskette, and list the programs that have been saved on it.

4. Have the student make a backup copy of a diskette.

5. Observe the student's knowledge of and skill in using the keyboard by having them key in a simple program (supplied by you) to the computer's memory. Pay special attention to their knowledge of the <--, -->, CTRL X, and CTRL C keys.

Ideally, these observations should happen under controlled test conditions--a performance test. However, if this is not possible, special care should be taken to ensure that the student can do all the above by herself/himself. You may wish to use a check sheet, while you observe them as they complete the learning activity.

Give the student a short written quiz on the new terminology covered in this activity. Randomly pick five of the terms and have them supply the definitions from recall. To be considered successful, they should be able to supply the definitions for four of these.

Computer Graphics Programming on an Apple II

GRAPHICS PROGRAMMING IN BASIC USING ALPHANUMERIC CHARACTERS

Introduction: This activity is the first in a series of activities that is designed to teach the student how to draw computer images by programming the computer in the BASIC language. This activity can be done on any Apple computer that uses Applesoft BASIC, the II+, IIe, IIc, or IIgs.

Enabling Objective: Before beginning this activity, the student should have completed Activity COG-003, How to Operate the Apple Computer. He or she should be able to turn the computer off and on, be familiar with the keyboard, be able to load and save programs to the disk drive, and be able to use the printer.

Performance Objective (5-10 days):

TASK/COMPETENCY--COG-004

The student should be able to write a computer graphics program using Applesoft variable names and the Applesoft BASIC commands HTAB, VTAB, HOME, LET, PRINT, REM, and END. This program will use alphanumerics to draw graphic images. The student should also be able to define the following terms associated with BASIC programming:

compiler	interpreter
BASIC (computer language)	machine language
source code	object code
RAM	ROM

CONDITIONS

The above tasks should be performed under the following conditions:

a) The student should be provided an Apple computer with a disk drive, monitor, and a printer and an Applesoft BASIC reference manual when writing the program.

b) Terms should be defined without the aid of a reference.

STANDARD OF PERFORMANCE

The above tasks should be performed to the following standards:

a) The graphics program should only incorporate alphanumeric characters and those Applesoft commands listed. It should run without error and draw the intended image.

b) The student should be able to supply a correct definition for four out of five terms. Definitions should be considered correct if they conform to definitions found in the current computer literature.

Selected Instructional Materials: An Apple II+, IIe, IIc, or IIgs computer with printer, monitor (preferably with color), and disk drive; an Applesoft reference manual; an Apple II utilities diskette or system master diskette; a formatted diskette, a piece of 40 x 24 unit graph paper; and a pencil.

Teacher References

Recommended

Books:

Apple Computer, Inc. (1983). *Applesoft BASIC programmer's reference manual.* Reading, MA: Addison-Wesley.

Apple Computer, Inc. (1986). *A touch of Applesoft BASIC.* Cupertino, CA: Author.

Software:

Apple IIgs System Disk, ProDOS User's Disk, or *Apple II System Master (DOS 3.3).*

Suggested

Book:

Bitter, G., & Severe, S. (1986). *Programming in BASIC for the Apple II series.* Reading, MA: Addison-Wesley.

Teacher Activities

1. Familiarize yourself with the Applesoft BASIC commands LET, PRINT, REM, VTAB, HTAB, and END.

2. Familiarize yourself with the rules for using variables in Applesoft BASIC--integer, real, and string (character) variables.

3. Explain to the student the purpose of a program, where it resides in the computer's memory, and why it is necessary to save it to the diskette frequently when it is being edited. It should also be explained why computers need compilers or interpreters and the difference between a high level language like BASIC and machine language. Stress the terms, source code, BASIC, interpreter, RAM, ROM, object code, and machine language.

4. Present the above commands and rules to the student, preferably using the demonstration method or a self instructional module. The following are some reminders:

   a. Cover the purpose of the "," and the ";" when using the PRINT command.

b. Show how the "?" is used for the PRINT command.

c. Show how to print characters and calculations.

d. After showing how commands can be used without line numbers, show how to use line numbers to cause the computer to defer execution of commands.

e. Stress that the LET command is optional, but good BASIC programming style requires its use. (the ANSI standard BASIC also requires it).

f. When presenting the rules for assigning variable names, be sure to remember to explain about "reserved words."

g. Explain that although a simple program will execute without an END statement, more complicated programs will not; therefore, students should form the habit of always using it.

h. Distinguish between the LET, PRINT, REM, and END, and the HTAB and VTAB statements. The former statements are standard BASIC commands, while the latter are extensions added by Apple to standard BASIC in Applesoft BASIC. If they write a program using HTAB and VTAB and later wish to convert the program for use on another computer, they will need to change these commands.

11. Familiarize yourself with the editing procedure using the escape and the I, J, K, and M keys or the escape and arrow keys (called escape mode). Demonstrate this procedure to your students. A demonstration is preferable, because this procedure is not easily understood, unless it is observed.

Student References

Recommended

Books:

Apple Computer, Inc. (1983). *Applesoft BASIC programmer's reference manual*. Reading, MA: Addison-Wesley.

Apple Computer, Inc. (1986). *A touch of Applesoft BASIC*. Cupertino, CA: Author.

Software:

Apple IIgs System Disk, ProDOS User's Disk, or *Apple II System Master (DOS 3.3)*.

Suggested

Book:

Bitter, G., & Severe, S. (1986). *Programming in BASIC for the Apple II series*. Reading, MA: Addison-Wesley.

Student Activities

1. Read about the REM, PRINT, LET, HTAB, VTAB, and END statements in one of the above references.

2. In the same references, read about numbering lines when writing a BASIC program and how to use variables. You should understand about integer, real, and string variables.

3. See your instructor to receive a demonstration or a self instructional module on the above topics. You can also work through the first two sections of the reference, *A Touch of Applesoft BASIC*.

4. Read about and then have your instructor demonstrate the escape mode method of editing a program on an Apple computer.

5. Before graphics techniques and graphics output devices became commonplace, programmers drew pictures on alphanumeric printers using alphanumeric characters. The following program shows how this was done. Key this program into your computer, RUN the program, edit if necessary, and SAVE it under the file name COG-004.1 on your diskette. **Be careful, spacing is important**.

[COG.004.1]

```
NEW

100 REM **   ACTIVITY FOUR    **
110 REM * COMPUTER GRAPHICS *
120 HOME
130 PRINT "             X             "
140 PRINT "           X   X           "
150 PRINT "         XX       XX       "
160 PRINT "    XX               XX    "
170 PRINT "XX                       XX"
180 PRINT "  X    MYSCHOOL     X  "
190 PRINT "  X                 X  "
200 PRINT "  X       XXX       X  "
210 PRINT "  X       X  X      X  "
220 PRINT "  X       X  X      X  "
230 PRINT "  X       X  X      X  "
240 END
```

6. Using the graph paper representing the Apple text screen (see Appendix A), draw the A/N school house you programmed above. Add a flagpole.

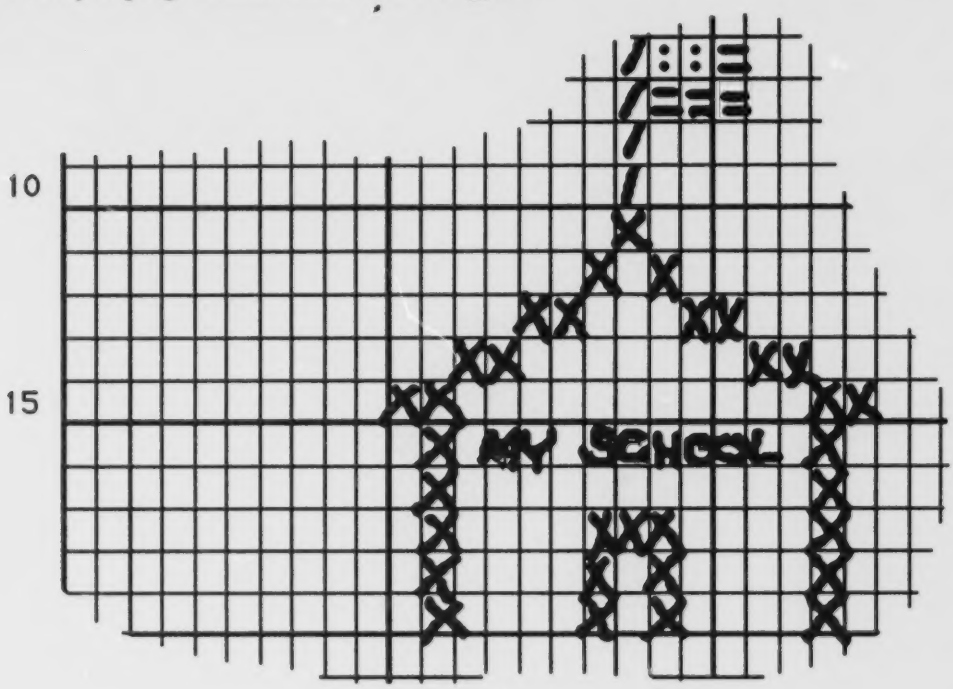


Figure 5--School with Flagpole

7. Now edit your school house program, adding the flagpole. SAVE this program under a different name (COG.004.2) to preserve the earlier version.

[COG.004.2]

8. One problem with these programs is that the drawing always appears in the upper left side of the screen. How can the image be moved to other positions on the screen? The following program shows how to use the HTAB and VTAB commands to do this. Change your earlier program by adding the new line numbers, RUN the new program, edit it if necessary, and SAVE it under a different name (COG.004.3).

[COG.004.3]

```
NEW

100 REM **    ACTIVITY FOUR    **
110 REM * COMPUTER GRAPHICS *
115 HOME
116 LET X = 14
118 LET Y = 5
120 VTAB Y
121 HTAB X
122 PRINT "              1 : : =          "
123 HTAB X
124 PRINT "              1= = =           "
125 HTAB X
126 PRINT "              1                "
127 HTAB X
128 PRINT "              1                "
129 HTAB X
130 PRINT "              X                "
135 HTAB X
140 PRINT "            X  X               "
145 HTAB X
150 PRINT "         XX      XX            "
155 HTAB X
160 PRINT "    XX              XX         "
165 HTAB X
170 PRINT "XX                      XX"
175 HTAB X
180 PRINT "   X   MY SCHOOL    X   "
185 HTAB X
190 PRINT "   X                X   "
195 HTAB X
200 PRINT "   X       XXX      X   "
205 HTAB X
210 PRINT "   X       X  X     X   "
215 HTAB X
220 PRINT "   X       X  X     X   "
225 HTAB X
230 PRINT "   X       X  X     X   "
240 END
```

9. Change the values in lines 116 and 118 to move the image around on the screen. Experiment with these values.

10. The following is a partial program that uses a different method of drawing the same image (the familiar schoolhouse), complete this program. Hint: it is easier, if you use the graph paper describing the Apple text screen.

[COG.004.4]

```
NEW

100 REM ** ACTIVITY FOUR **
110 REM * COMPUTER GRAPHICS *
120 HOME
130 LET A$ = "1 : : ="
140 LET B$ = "1= = ="
150 LET C$ = "1"
160 LET D$ = "X"
170 LET X = 10
180 LET Y = 4
190 HTAB X + 7: VTAB Y + 1
200 PRINT A$
210 HTAB X + 7: VTAB Y + 2
220 PRINT B$
230 HTAB X + 7: VTAB Y + 3
240 PRINT C$
250 HTAB X + 7: VTAB Y + 4
260 PRINT C$
270 HTAB X + 7: VTAB Y + 5
280 PRINT D$
290 HTAB X + 6: VTAB Y + 6
300 PRINT D$;" ";D$
999 END
```

11. Using the Apple text screen graph paper draw an image of your choice. Next, write a program using the PRINT command that will draw your image on the display screen. The program you write should be like the programs in the previous parts of this activity.

Evaluation Criteria: To evaluate the student's success in completing this activity, the teacher should give the student a two part test. The first part should be a performance test where the student is given a simple drawing on the Apple text screen graph paper. The student should be instructed to write a program that will draw the image using the PRINT command and alphanumeric characters. The second part should be a recall test on which the student is asked to define five randomly chosen terms which were covered in this activity.

To be considered successful, the student's program should run on the computer and draw the intended image. On the second part, the student should be able to provide an acceptable definition for four of the five terms.

Computer Graphics Programming on an Apple II

GRAPHICS PROGRAMMING IN BASIC USING APPLESOFT LOW-RESOLUTION COMMANDS

Introduction: This activity is the second in a series of activities that is designed to teach the student how to draw computer images by programming the computer in the BASIC language. This activity can be done on any Apple computer that uses Applesoft BASIC, the II+, IIe, IIc, or IIgs.

Enabling Objective: Before beginning this activity, the student should have completed Activity COG--004, Graphics Programming in BASIC Using Alphanumeric Characters. He or she should have demonstrated mastery of the objectives found in that activity before beginning this one.

Performance Objective (5-10 days):

TASK/COMPETENCY--COG-005

The student should be able to write a low resolution graphics program using the Applesoft BASIC commands GOTO, GR, PLOT, HLIN, VLIN, COLOR, INPUT, READ, and DATA. He or she should also be able to define the following computer graphic terms:

pixel
resolution
Cartesian coordinates

CONDITIONS

The above tasks should be performed under the following conditions:

a) The student should be provided an Apple computer with a disk drive, monitor, and a printer and an Applesoft BASIC reference manual.

b) Terms should be defined without the aid of a reference.

STANDARDS OF PERFORMANCE

The above tasks should be performed to the following standards:

a) The graphics program should only incorporate low-resolution graphics commands and those other Applesoft commands listed above or covered in COG-004. It should run without error (except for an out of data error) and draw the intended image.

b) The student should be able to supply a correct definition for all of the terms. Definitions should be considered correct if they conform to definitions found in the current computer literature.

Selected Instructional Materials: An Apple II+, IIe, IIc, or IIgs computer with printer, monitor (preferably with color), and disk drive; an Applesoft reference manual; an Apple II utilities diskette or system master diskette; a formatted diskette; the *Computer Graphics Curriculum Project Master Diskette*; a piece of graph paper representing the Apple low-resolution graphics screen; and a pencil.

Teacher References

Recommended

Books:

Apple Computer, Inc. (1983). *Applesoft BASIC programmer's reference manual*. Reading, MA: Addison-Wesley.

Apple Computer, Inc. (1986). *A touch of Applesoft BASIC*. Cupertino, CA: Author.

Software:

Apple IIgs System Disk, ProDOS User's Disk, or *Apple II System Master (DOS 3.3).*

Suggested

Book:

Bitter, G., & Severe, S. (1986). *Programming in BASIC for the Apple II series*. Reading, MA: Addison-Wesley.

Teacher Activities

1. Familiarize yourself with the Applesoft BASIC commands GOTO, GR, PLOT, HLIN, VLIN, COLOR, INPUT, READ, and DATA.

2. Explain to the student the meaning of the terms pixel and resolution. Also explain the Cartesian coordinate system. Note: explain to the student how the Apple screen is set up differently than most other screens used for graphics.

3. Present the above commands and rules to the student, preferably using the demonstration method or a self instructional module. The following are some reminders:

   a. Demonstrate the immediate mode of command execution and how it can be used to input data from the keyboard. When you do, also explain that the RUN command should be avoided when working in this way because it causes variables to be reset to zero.

   b. Show how the TEXT command can be used to return to the text page after using the graphics page.

   c. Show how the READ and DATA commands can be used to handle all types of data; but, in order to do so, the variables and data must match.

d. Distinguish between the GOTO, READ, INPUT, AND DATA Applesoft commands and the GR, PLOT, COLOR, HLIN, and VLIN Applesoft commands. The former commands are standard BASIC commands, while the latter are extensions added by Apple to standard BASIC. If students write a program using HLIN and VLIN and later wish to convert the program for use on another computer, they will need to change these commands.

4. Familiarize yourself with the "LoRes Editor" program found on the *Computer Graphics Curriculum Project Master Diskette*. Be prepared to help students draw with the program.

Student References

Recommended

Books:

Apple Computer, Inc. (1983). *Applesoft BASIC programmer's reference manual*. Reading, MA: Addison-Wesley.

Apple Computer, Inc. (1986). *A touch of Applesoft BASIC*. Cupertino, CA: Author.

Software:

Apple IIgs System Disk, ProDOS User's Disk, or *Apple II System Master (DOS 3.3).*

Suggested

Book:

Bitter, G., & Severe, S. (1986). *Programming in BASIC for the Apple II series*. Reading, MA: Addison-Wesley.

Student Activities

1. Read about the Applesoft BASIC commands GOTO, GR, PLOT, HLIN, VLIN, COLOR, INPUT, READ, and DATA in one of the above references.

2. See your instructor to receive a demonstration or a self instructional module on the above topics.

3. Read about and then have your instructor demonstrate the difference between immediate mode and deferred mode command execution.

4. Review the READ and DATA commands. Using these commands, provides you with another way to supply data to your program. This is demonstrated in the following program. Boot a diskette with DOS 3.3 or use your ProDos utility diskette to program in BASIC. Then, enter the following program into your computer:

[COG.005.1]

```
10 REM **      ACTIVITY FIVE     **
20 REM * COMPUTER GRAPHICS *
30 HOME
40 READ A$,B$,C$,D$,E$,F$,G$
50 PRINT A$:PRINT B$
60 PRINT C$:PRINT D$
70 PRINT E$:PRINT F$
80 PRINT G$
100 DATA *     * * *
110 DATA *     * * *
120 DATA *     * * *
130 DATA *****  * *
140 DATA *     * * *
150 DATA *     * *
160 DATA *     * * *
170 END
```

RUN your program. Make corrections, if needed. Don't hesitate to play with it by changing lines. Be sure to SAVE your finished program to diskette.

Besides the use of the READ and DATA commands, something else is different about the above program. Notice that on lines 50, 60, and 70 there are two commands on each of the lines. Read about this in one of the above references. Look under using the : (colon) for listing multiple instructions on a line.

5. Review the GOTO command. Then, change the above program by DELETING lines 40-80 and adding the following lines:

```
40 READ A$
50 PRINT A$
60 GOTO 40
```

When finished, your program should look as the following does. However, it will do the same thing as it did before you changed it, only now with fewer commands. Note: This program does cause an OUT OF DATA error message to occur. We will correct that in subsequent activities.

[COG.005.2]

```
10 REM **  ACTIVITY FIVE  **
20 REM * COMPUTER GRAPHICS *
30 HOME
40 READ A$
50 PRINT A$
60 GOTO 40
100 DATA *     * * *
110 DATA *     * * *
```

```
120 DATA *     * * *
130 DATA *****  * *
140 DATA *     * * *
150 DATA *     * *
160 DATA *     * * *
170 END
```

RUN, debug, and SAVE this program with a unique name (COG.005.2).

6. Review the GR, COLOR, and PLOT commands. Then try the following program:

[COG.005.3]

```
10 GR
20 COLOR = 1
30 PLOT 20,20
40 END
```

This demonstrates the simplest form of low-resolution graphics on an Apple computer. We turned on a pixel (what does this term mean?) in the middle of the screen.

Remember the demonstration on immediate mode operation of your computer? Use it now to return to a full page of text by keying in the following (remember to leave off the line numbers):

```
TEXT
HOME
LIST
```

7. Change lines 20 and 30 so that your program looks like the following:

COG.005.3.1]

```
10 GR
20 COLOR = C
30 PLOT X,Y
40 END
```

Look your program over carefully after keying it in. If it is okay, then do the following in immediate mode:

```
LET C= 2
LET X= 10
LET Y= 10
GOTO 10
```

Notice that you should use the GOTO 10 rather than RUN. RUN causes all variables to return to zero. Therefore, it would cancel the values you have just assigned to C, X, and Y. Again, you can return to a full page of text by:

```
TEXT
HOME
```

Or, you can continue as is, with just four lines of text at the bottom of the graphics screen.

Repeat the process of assigning values to C, X, and Y in immediate mode to obtain a feel for low-resolution graphics and the commands GR, COLOR, and PLOT. If you would like to keep the previous plot, use GOTO 20 instead of GOTO 10--reason: the GR command (at line 10) clears the screen. You don't need to change all of the values each time. Here are some examples:

```
LET C = 12
LET X = 10
LET Y = 10
GOTO 10
```

```
LET X= 20
GOTO 20
```

```
LET C= 8
LET Y= 30
GOTO 20
```

Color = 0,16,32 etc. will cause a black pixel to be turned on. The result will be that you will not see anything because of the black background on your display screen. That is a good way to erase something!!!

8. Previously, in parts 5 & 6, the DATA commands were used to supply string data to the program. The following program uses the DATA commands to supply numeric values to the program. Therefore, the READ commands are followed by numeric variables rather than string variables and the data statements have numeric data rather than string data. Key the following program into your computer--be sure to key in NEW first:

[COG.005.4]

```
10 REM **     ACTIVITY FIVE     **
20 REM * COMPUTER GRAPHICS *
30 GR : COLOR = 9
40 READ X,Y
50 PLOT X,Y
60 GOTO 40
100  DATA  1,6,1,7,1,8,2,4,2,5,2,9,2,10,3,3,3,11,4,2,4,12,5
110  DATA  2,5,12,6,1,6,13,7,1,7,13,8,1,8,13,9,2,9
120  DATA  12,10,2,10,12,11,3,11,11,12,4,12,5,12,9,12
130  DATA  10,13,6,13,7,13,8,5,6,9,6,7,7,7,8,5,10,6,10
140 DATA 7,10,8,10,9,10
150 END
```

This program does end with an OUT OF DATA error message. You will be shown how to prevent it in a later activity.

RUN the program. If the image that is drawn is not a face, check the values in your DATA commands. It is very easy to miss something when you key in this many numbers. When the program runs as it should, SAVE it to diskette.

9. Read again about the Applesoft commands HLIN and VLIN. Then try HLIN with the following program:

[COG.005.5]

```
10 GR
20 COLOR = 1
30 HLIN 0,39 AT 10
40 END
```

Change line 30 to read 30 HLIN 10,29 AT 10. Then to 30 HLIN 10,29 AT 20.

[COG.005.5.1]

[COG.005.5.2]

Now change to the following and give VLIN a try:

[COG.005.5.3]

```
10 GR
20 COLOR = 5
30 VLIN 0,39 AT 10
40 END
```

Play around with this program. Change lines and values to get a feel for VLIN. Try adding lines to draw both HLIN's and VLIN's or more than one VLIN at a time.

Try changing the program to look like this: This version will end with an ILLEGAL QUANTITY ERROR IN 30 message. Do not be concerned.

[COG.005.5.4]

```
10 GR
15 LET C= C+1
20 COLOR = C
30 HLIN 0,39 AT C
40 GOTO 15
50 END
```

10. Read again about the Applesoft command INPUT. Then change the above program to use the command:

[COG.005.5.5]

```
10 GR
15 INPUT C
20 COLOR = C
30 HLIN 0,39 AT C
40 GOTO 15
50 END
```

Now, play with the program. Remember, that you will need to input a number between 0-39 from the keyboard each time. To stop the program use CTRL C before you input a value for the last time.

11. Using the graph paper representing the Apple low-resolution screen (see Appendix A), reproduce the drawing below. Now write an Applesoft BASIC program that will draw the object on the screen.

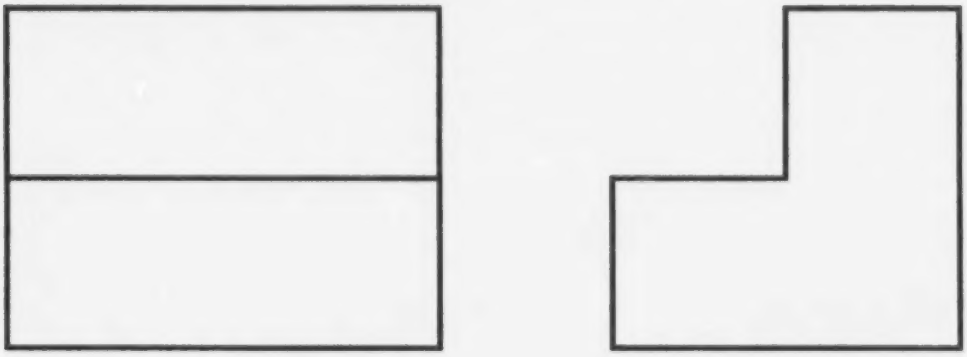


Figure 6--Two View Drawing

12. Obtain a copy of the program LoRes Screen Editor from your instructor. LOAD the program and RUN it. This program makes liberal use of the commands you have used in this activity (as well many more). LIST it and look it over, but don't feel bad about not understanding it. MOST IMPORTANTLY, have fun drawing with it. Below is a reprint of the instructions that go with this program:

    LORES SCREEN EDITOR

    CURSOR MOVEMENT:
    I KEY = UP
    M KEY = DOWN
    J KEY = LEFT
    K KEY = RIGHT
    U KEY = UPPER LEFT
    O KEY = UPPER RIGHT
    , KEY = LOWER RIGHT
    N KEY = LOWER LEFT
    L = LOAD PICTURE
    S = SAVE PICTURE, Q = QUIT,
    E = ERASE,C = CHANGE COLOR,
    P = DRAW AT THIS LOCATION
    PRESS <CR> TO BEGIN

    After the program has been RUN, the instructions will appear. When you are ready to begin, you press the carriage return <CR>. When you do, a blinking LoRes block (pixel) will appear on the screen. You can move this blinking cursor by pressing the I,M,J,K,U,O,N, or comma keys. If you would like to leave the pixel in that particular location lighted with the present color, you press the "P" key. Pressing one of the movement keys and the "P" keys alternately will allow you to draw a line.

Evaluation Criteria: To evaluate the student's success in completing this activity, the teacher should give the student a two part test. The first part should be a performance test where the student is given a simple drawing on the Apple low-resolution graph paper. The student should be instructed to write a program that will draw the image using the commands learned in this activity and activity COG-004. The second part should be a recall test with the terms which were covered in this activity.

To be considered successful, the student's program should run on the computer and draw the intended image. On the second part, the student should be able to provide an acceptable definition for all of the terms.

Computer Graphics Programming on an Apple II

GRAPHICS PROGRAMMING IN BASIC USING BRANCHING AND LOOPING COMMANDS

Introduction: This activity is the third in a series of activities that is designed to teach the student how to draw computer images by programming an Apple computer with the BASIC language. This activity can be done on any Apple computer that uses Applesoft BASIC, the II+, IIe, IIc, or IIgs.

Enabling Objective: Before beginning this activity, the student should have completed Activity COG-005, Graphics Programming in BASIC Using Applesoft Low-Resolution Commands. He or she should have demonstrated mastery of the objectives found in that activity before beginning this one.

Performance Objective (5-10 days):

TASK/COMPETENCY--COG-006

The student should be able to write a low resolution graphics program using the Applesoft BASIC commands GET, IF...THEN, FOR...TO and NEXT, and ON...GOTO. He or she should also be able to define the following terms associated with graphics programming:

fatal error
aspect ratio
interactive
batch mode

CONDITIONS

The above tasks should be performed under the following conditions:

a) The student should be provided an Apple computer with a disk drive, monitor, and a printer and an Applesoft BASIC reference manual when writing a graphics program.

b) The student should be expected to recall the above terms without the aid of a reference.

STANDARD OF PERFORMANCE

The above tasks should be performed to the following standards:

a) The graphics program should only incorporate low-resolution graphics commands and those other Applesoft commands listed above or covered in COG-004 or COG-005 activities. It should run without error and draw the intended image.

b) The student should be able to supply a correct definition for all of the terms. Definitions should be considered correct if they conform to definitions found in the current computer literature.

Selected Instructional Materials: An Apple II+, IIe, IIc, or IIgs computer with printer, monitor (preferably with color), and disk drive; an Applesoft reference manual; an Apple II utilities diskette or system master diskette; a formatted diskette; the *Computer Graphics Curriculum Project Master Diskette*; a piece graph paper representing the Apple low-resolution graphics screen; and a pencil.

Teacher References

Recommended

Books:	Apple Computer, Inc. (1983). *Applesoft BASIC programmer's reference manual.* Reading, MA: Addison-Wesley.
	Apple Computer, Inc. (1986). *A touch of Applesoft BASIC.* Cupertino, CA: Author.
Software:	*Apple IIgs System Disk, ProDOS User's Disk,* or *Apple II System Master (DOS 3.3).*

Suggested

Book:	Bitter, G., & Severe, S. (1986). *Programming in BASIC for the Apple II series.* Reading, MA: Addison-Wesley.

Teacher Activities

1. Familiarize yourself with the Applesoft BASIC branching and looping commands IF...THEN, FOR...TO and NEXT, and ON...GOTO.

2. Explain to the student the meaning of the terms fatal error, aspect ratio, interactive, and batch mode. Also, demonstrate how the aspect ratio of a display device can be measured and computed using the computer as a calculator.

3. Present the above commands to the student, preferably using the demonstration method or a self instructional module. The following are some reminders:

   a. When explaining the IF...THEN command be sure to emphasize all of the logical and relational operators. Give the student some written exercises to work with involving logical and relational operators.

   b. Make sure the student understands the arithmetic operators and operations before she/he begins Part 6 of the student activity. Again, some written problems might be a good exercise for the student.

c. When explaining the FOR...TO and NEXT commands, don't forget to explain how they can be nested and the proper procedure for nesting. Also explain the FOR...TO...STEP version of this command.

d. Show the student how the Applesoft GET command can be used in place of the INPUT command for some types of input.

4. Run and study all of the programs found in the student activity. Be sure you understand the logic of the programs and all of the commands used in them. If time permits, work through the student activities.

Student References

Recommended

Books:

Apple Computer, Inc. (1983). *Applesoft BASIC programmer's reference manual*. Reading, MA: Addison-Wesley.

Apple Computer, Inc. (1986). *A touch of Applesoft BASIC*. Cupertino, CA: Author.

Software:

Apple IIgs System Disk, ProDOS User's Disk, or *Apple II System Master (DOS 3.3).*

Suggested

Book:

Bitter, G., & Severe, S. (1986). *Programming in BASIC for the Apple II series*. Reading, MA: Addison-Wesley.

Student Activities:

1. Read about the Applesoft BASIC command IF...THEN in one of the above references.

In the previous activity, COG-005, you were told that you would be shown how to correct the OUT OF DATA error caused by the following program.

[COG.005.2]

```
10 REM **    ACTIVITY FIVE     **
20 REM * COMPUTER GRAPHICS *
30 HOME
40 READ A$
50 PRINT A$
60 GOTO 40
```

```
100 DATA *    * * *
110 DATA *    * * *
120 DATA *    * * *
130 DATA ***** * *
140 DATA *    * * *
150 DATA *    * *
160 DATA *    * * *
170 END
```

Add line 35 and change line 60 as shown below, to correct the OUT OF DATA error:

```
35 LET N = N + 1
60 IF N < 7 GOTO 35
```

You may also want to change the REM statement at line 10.

The program should now look like this:

[COG.006.1]

```
10 REM **  ACTIVITY  SIX   **
20 REM * COMPUTER GRAPHICS *
30 HOME
35 LET N = N + 1
40 READ A$
50 PRINT A$
60 IF N < 7 GOTO 35
100 DATA *    * * *
110 DATA *    * * *
120 DATA *    * * *
130 DATA ***** * *
140 DATA *    * * *
150 DATA *    * *
160 DATA *    * * *
170 END
```

RUN, debug, and SAVE this program with a unique name (COG.006.1).

In the above program, the LET command at line 35 acts as a counter. When the program begins, N is equal to zero (remember the RUN command causes all variables to be reset to zero). The first time line 35 is executed, N becomes 1, the second time, N becomes 2, etc. The program continues to return to line 35 while N is less than 7. This is the function of the IF..THEN statement at line 60. Therefore, the READ and PRINT statements are executed seven times--just the correct number to read all the DATA in this program.

2. Read about the Applesoft BASIC command FOR...TO and NEXT in one of the above references. Then change lines 35 and 60 in the above program to utilize the FOR...TO and NEXT commands.

```
35 FOR N = 1 TO 7
60 NEXT N
```

When changed, the program should look like the following. This version will also eliminate the OUT OF DATA error message.

[COG.006.2]

```
10 REM **   ACTIVITY   SIX    **
20 REM * COMPUTER GRAPHICS *
30 HOME
35 FOR N = 1 TO 7
40 READ A$
50 PRINT A$
60 NEXT N
100 DATA *    * * *
110 DATA *    * * *
120 DATA *    * * *
130 DATA ***** * *
140 DATA *    * * *
150 DATA *    * *
160 DATA *    * * *
170 END
```

RUN, debug, and SAVE this program with a unique name (COG.006.2).

3. The following program uses a FOR...TO and NEXT loop (lines 90-120) to draw sixteen low-resolution lines on the screen, one in each of the sixteen available colors (remember that one of the colors is black, the same as the background). Notice how the variable used to count the loops, N, is also used within the loop to change the colors.

   A second loop (lines 160-180) is used to delay the program so the bars can be seen before the screen is erased and the TEXT page is displayed. Key it in and RUN it.

[COG.006.3]

```
10 REM **   ACTIVITY    SIX   **
20 REM * COMPUTER GRAPHICS *
30 REM ************************
40 REM *  MAIN  LOOP          *
50 REM *  DRAWS  SIXTEEN      *
60 REM *  COLOR  BARS         *
70 REM ************************
80 GR
90 FOR N = 0 TO 15
100 COLOR = N
110 VLIN 10,30 AT N+12
120 NEXT N
130 REM ************************
140 REM *   A  DELAY  LOOP     *
150 REM ************************
160 FOR M = 1 TO 5000
170 PRINT M
180 NEXT M
190 REM ************************
200 REM * CLEARS THE SCREEN *
210 REM ************************
```

```
220 TEXT
230 HOME
240 END
```

After you RUN this program, SAVE it. Then delete lines 160, 170, & 180 (DEL 160,180). Now run the program to see why you need the delay loop. Restore the original version of the program by keying the following:

```
NEW
LOAD COG.006 .3 (or whatever name you saved it under).
```

Delete line 170 and then run the program to see how the PRINT command increases the delay.

Experiment with the loops by making these changes:

a) Move the 16 bars to other screen locations.

b) Change line 90 to FOR N = 0 TO 31 and change the N+12 in line 110 to N+4 (can you explain the results?).

c) Change the value 5000 in the second loop to another value.

d) Change the VLIN in line 110 to an HLIN.

4. Read about nested loops in one of the above references. Then change the above program so that it looks like the following:

[COG.006.4]

```
10 REM **   ACTIVITY  SIX    **
20 REM * COMPUTER GRAPHICS *
30 REM ************************
40 REM * PAINTS  THE  SCREEN *
50 REM ************************
60 GR
70 COLOR = 9
80 FOR H = 0 TO 39
90 FOR V = 0 TO 39
100 PLOT H,V
110 NEXT V
120 NEXT H
130 REM ************************
140 REM *  A  DELAY  LOOP       *
150 REM ************************
160 FOR M = 1 TO 5000
170 PRINT M
180 NEXT M
190 REM ************************
200 REM * CLEARS  THE  SCREEN *
```

60

```
210 REM ************************
220 TEXT
230 HOME
240 END
```

RUN, debug, and SAVE this version of the program with a unique name (COG.006.4).

Why do the loops count from 0 to 39? Change them and see what happens.

Watch how the screen is painted, then reverse the loops by moving line 80 to 95 and 120 to 105, like so:

[COG.006.4.1]

```
60 GR
70 COLOR = 9
90 FOR V = 0 TO 39
95 FOR H = 0 TO 39
100 PLOT H,V
105 NEXT H
110 NEXT V
```

RUN this version. Would this have worked (run without an error message) had line 120 not been moved to 105? Why?

Rewrite this program using the HLIN or VLIN command to paint the screen. Experiment with different patterns...how about a basket weave?

Read about the FOR, TO, STEP command in one of the above references. Now, change the order that the screen is painted by changing lines 80 & 90 as follows: Notice that lines 95 and 105 have been eliminated.

[COG.006.4.2]

```
60 GR
70 COLOR = 9
80 FOR H = 39 TO 0 STEP -1
90 FOR V = 39 TO 0 STEP -1
100 PLOT H,V
110 NEXT V
120 NEXT H
```

5. The following program written by Mr. W.J. Masalski utilizes alphanumeric (PRINT command) graphics and the FOR,TO AND NEXT commands to do some simple animation. In this program an "O" represents a falling egg which strikes the ground and breaks "X". The illusion of the falling egg is created by printing the "O" in a series of different locations. After each "O" is printed, there is a pause and then the "O" is erased before the next one is printed. It is erased by printing over it with a blank space (" "). Key this program into your computer.

[COG.006.5]

```
10 REM **   ACTIVITY  SIX    **
20 REM * COMPUTER GRAPHICS *
30 REM ************************
```

```
40 REM * FALLING EGG          *
50 REM * BY: W. J. MASALSKI   *
60 REM *************************
70 REM * DRAWS THE GROUND    *
80 REM * WITH .....'S        *
90 REM *************************
100 HOME
110 FOR X = 1 TO 40
120 VTAB 20
130 HTAB X
140 PRINT "."
150 NEXT X
160 REM ***********************
170 REM * DRAWS THE FALL-    *
180 REM * ING EGG            *
190 REM ***********************
200 FOR A = 1 TO 19
210 HTAB A
220 VTAB A
230 PRINT "O"
240 FOR C = 1 TO 150: NEXT C
250 HTAB A
260 VTAB A
270 PRINT " "
280 NEXT A
290 REM ***********************
300 REM * EGG STRIKING        *
310 REM * THE GROUND AND     *
320 REM * EXPLODING          *
330 REM ***********************
340 HTAB A
350 VTAB A
360 PRINT "X"
370 HTAB 19
380 VTAB 19
390 PRINT "' '"
400 END
```

RUN, debug, and SAVE this program with a unique name (COG.006.5).

The loop at line 240 delays the action so that the egg doesn't fall so fast that you can't see it. Delete this line and run the program to see why it is necessary to do this. After you run it, replace line 240. Change the value of 150 in this loop to another value, 25, 200, . . , to see the effect of different time lapses.

Next, delete line 270 and RUN the program to see why it is necessary to erase the "O" each time.

Feel free to change the program as much as you like to add your own touch....make the breaking more dramatic by adding additional "' '", or make the egg bounce instead of break. *Be creative!!!*

6. In the next part of this activity, you will be designing a program that is designed to be interactive. This means that the program expects the person running it to input data, thereby giving direction to the program as it is running.

   The goal of our program will be to draw a square wherever the user would like. It will ask the user to input X,Y coordinates describing the location of the corner of a square and it will then draw a square positioned at that location. Start by keying in the following program. Notice that the program prints directions for the user, asks for the user to input coordinates, and then has a routine for drawing the square with the HLIN and VLIN commands.

   You may want to read again about the INPUT command before you begin. Notice that in this program the INPUT command is used with a prompting message.

   [COG.006.6]

```
10 REM **    ACTIVITY   SIX    **
20 REM * COMPUTER GRAPHICS *
30 HOME
40 PRINT "THIS PROGRAM DRAWS"
50 PRINT "A SQUARE AT A"
60 PRINT "LOCATION OF YOUR"
70 PRINT "CHOICE"
80 PRINT:PRINT
90 PRINT "YOU WILL NEED TO"
100 PRINT "INPUT THE LOCATION"
110 PRINT "WHEN ASKED. YOUR"
120 PRINT "COORDINATES SHOULD"
130 PRINT "HAVE VALUES BE-"
140 PRINT "TWEEN 0-35."
150 REM ************************
160 REM * A  DELAY  LOOP        *
170 REM ************************
180 FOR N = 1 TO 9000:NEXT N
190 REM ************************
200 REM * ASK FOR USER INPUT*
210 REM ************************
220 INPUT "X (0--35) = ?";X
230 INPUT "Y (0--35) = ?";Y
1000 REM **********************
1010 REM * ROUTINE TO DRAW  *
1020 REM * SQUARE           *
1030 REM **********************
1040 GR
1050 COLOR = 2
1060 HLIN X,X+4 AT Y
1070 HLIN X,X+4 AT Y+4
1080 VLIN Y,Y+4 AT X
1090 VLIN Y,Y+4 AT X+4
1100 GOTO 220
1110 END
```

RUN, debug, and SAVE this program with a unique name (COG.006.6). **To exit the program you must use CRTL C while pressing the RETURN key.**

When you RUN this program, you may notice that the squares aren't quite square. This will be true on most display screens, even though in lines 1060 thru 1090 the starting and stopping of the HLIN and VLIN commands were arranged to make the squares 4 pixels by 4 pixels. Yes, you guessed right--the pixels are not square. Therefore, you must make adjustments to these commands based on the aspect ratio of your display screen--display screens are not alike, they come in all sizes and aspect ratios. Before you make these adjustments, look up the definition of aspect ratio in a dictionary or have your instructor explain it.

Read about arithmetic operators and the precedence of operators in one of the above references. If you don't understand what is being done in lines 1060 thru 1090, ask your instructor for an explanation.

RUN the program again and draw at least one block (we won't call them squares yet). Take a ruler, place it on the screen as best you can, and measure the height and width of the block. You can now compute the aspect ratio of the pixels drawn on your display screen by the following formula:

ASPECT RATIO = HEIGHT/WIDTH

Let the computer do the calculations for you. For example, if your block measures 5/8 inch high by 1 inch wide, then divide 5/8 by 1 by keying in the following.

PRINT (5/8) / 1 or just PRINT 5/8

Your answer, .625, will appear on the screen. This is your aspect ratio and the value you will use to make adjustments in your HLIN and VLIN commands. To make the necessary adjustments to your commands, you should multiply the aspect ratio, .625, times the X dimension of your block--.625 x 4 pixels = 2.5 pixels. This will give you a square with a dimension approximately the same as the original block's height (1 inch). There is only one problem. The HLIN and VLIN commands don't recognize fractions. Therefore, your choices are to draw a block 2 x 4 or 3 x 4, if you choose to draw an approximate square using the height dimension.

You can also choose to use the width dimension of the original block as the dimension for your square. To do this, take the reciprocal of .625 (1/.625 = 1.6) and multiply it times the Y dimension of your block--1.6 x 4 pixels = 6.4 pixels. Since the fraction is not as large, this may be the best choice. If you choose to adjust the Y values rather than the X values, your square will have a dimension approximately equal to the width (1 and 1/8 inch) of the original block. You would then draw your blocks 4 x 6 pixels.

Change the commands in lines 1060, 1070, 1080, and 1090 to make the above adjustments. Try all of them, before you decide which you like best.

For a block 2 pixels x 4 pixels, make these changes:

[COG.006.6.1]

```
1060 HLIN X,X + 2 AT Y
1070 HLIN X,X + 2 AT Y + 4
1080 VLIN Y,Y + 4 AT X
1090 VLIN Y,Y + 4 AT X + 2
```

64

For a block 3 pixels x 4 pixels, make these changes:

[COG.006.6.2]

```
1060 HLIN X,X + 3 AT Y
1070 HLIN X,X + 3 AT Y + 4
1080 VLIN Y,Y + 4 AT X
1090 VLIN Y,Y + 4 AT X + 3
```

For a block 4 pixels x 6 pixels, make these changes:

[COG.006.6.3]

```
1060 HLIN X,X + 4 AT Y
1070 HLIN X,X + 4 AT Y + 6
1080 VLIN Y,Y + 6 AT X
1090 VLIN Y,Y + 6 AT X + 4
```

These correction factors are offered as examples only. Your display device may have a different aspect ratio. If so, none of these examples will be suitable, and you will need to compute a ratio based on your device's measurements.

You will find that this type of adjustment will be necessary when drawing graphics with any microcomputer that does not come equipped with a standard display device or with a microcomputer that can be purchased with optional standard display devices which have screens with different aspect ratios.

Even with the above correction, the original program may still need further refinements. As it now works, you can only have one square at a time on the screen. Change this by moving the GR command from line 1040 to line 215.

[COG.006.6.4]

7. It is also inconvenient to use the CTRL C--RETURN each time you wish to exit this program. Read about the ON...GOTO command and the GET command in one of the above references. Then, to allow the user to exit more conveniently, DELete lines 1100 and 1110 and add the following.

[COG.006.6.5]

```
1100 REM **********************
1110 REM *    EXIT ROUTINE     *
1120 REM **********************
1130 PRINT "DO YOU WANT TO"
1140 PRINT "1) QUIT OR 2) CONTINUE?"
1150 GET A
1160 ON A GOTO 1170,220
1170 TEXT
1180 HOME
1190 END
```

8. The square drawing program can be refined even further. It works well as long as the user is careful and inputs the correct values. But if he or she doesn't, the program crashes (it is a fatal error). There are some ways to help prevent that from happening. If you need to, go back and read about the IF..THEN command in one of the above references. Then, add the following lines to your program. These will help to check the user's input:

[COG.006.6.6]

```
222 IF X < 0 OR X > 35 GOTO 234
232 IF Y < 36 GOTO 1000
234 PRINT "PLEASE KEEP YOUR"
236 PRINT "INPUT BETWEEN 0 & 35"
238 GOTO 220
```

Now, if the user inputs a number for the X coordinate that is less than zero or greater than 35 (a number that is not acceptable) the program will GOTO line 234 and print the message shown there. After the message is printed, the program will GOTO line 220 where a new X coordinate is asked for and accepted by the INPUT command. However, if the X coordinate is okay--falls between 0 and 35--the IF statement is false and nothing happens at line 222. The program then executes line 230.

The same type of check is made on the Y coordinate at line 232. However, this time the program checks to see if the Y coordinate is acceptable. If it is, the program will GOTO line 1000 and draw the square, avoiding those lines which admonish the user and redirect the program for a correct input.

When finished, the part of your program from lines 190 to 1000 should look like this:

[COG.006.6.6]

```
190 REM ************************
200 REM * ASK FOR USER INPUT*
210 REM ************************
215 GR
220 INPUT "X (0--35) = ?";X
222 IF X < 0 OR X > 35 GOTO 234
230 INPUT "Y (0--35) = ?";Y
232 IF Y < 36 GOTO 1000
234 PRINT "PLEASE KEEP YOUR"
236 PRINT "INPUT BETWEEN 0 & 35"
238 GOTO 220

1000 REM ************************
1010 REM * ROUTINE TO DRAW   *
1020 . . . .
```

9. The preceding routine checks the user's input for numbers out of range. However, it does not check for those inputs which are not numbers. For example, a user mistakenly hits the wrong key and an "A' is sent to the program. If this happens, a fatal error will occur. All that can be done is to RUN the program again. If you would like to learn how to prevent this from occurring, read about the ONERR....GOTO command in one of the above references. As an optional exercise, change the above program to use the ONERR...GOTO command to do a better job of checking the user's input.

10. Develop an interactive program of your own. Keep it simple, but have the user make some input and then have the program react to that input in some predictable way. Try to incorporate those new commands you have learned in this activity.

Evaluation Criteria: To evaluate the student's success in completing this activity, the teacher should give the student a two part test. The first part should be a performance test where the student is required to write a computer graphics program incorporating the GET, FOR...TO, NEXT, IF...THEN, and ON...GOTO commands. The student should be instructed to use only those commands covered in this activity or those covered in activities COG-004 and COG-005. The second part should be a recall test requiring the student to define the terms which were covered in this activity.

To be considered successful, the student's program should run on the computer without error and draw the intended image. On the second part, the student should be able to provide an acceptable definition for all of the terms. A term will be considered acceptable if it agrees with definitions found in current computer literature.

Computer Graphics Programming on an Apple II

GRAPHICS PROGRAMMING IN BASIC USING APPLESOFT HIGH-RESOLUTION COMMANDS

Introduction: This activity is the fourth in a series of activities that is designed to teach the student how to draw computer images by programming an Apple computer with the BASIC language. This activity can be done on any Apple computer that uses Applesoft BASIC, the II+, IIe, IIc, or IIgs.

Enabling Objective: Before beginning this activity, the student should have completed Activity COG-006, Graphics Programming in BASIC Using Branching and Looping Commands. He or she should have demonstrated mastery of the objectives found in that activity before beginning this one.

Performance Objective (5-10 days):

TASK/COMPETENCY--COG-007

The student should be able to write a high-resolution graphics program using the Applesoft BASIC commands HGR, HCOLOR, HPLOT, PEEK, POKE, and CALL. He or she should also be able to define the following terms associated with graphics programming:

frame buffer
display buffer
bit map
high-resolution graphics display buffer

CONDITIONS

The above tasks should be performed under the following conditions:

a) The student should be provided an Apple computer with a disk drive, monitor, and a printer and an Applesoft BASIC reference manual when writing the graphics program.

b) The student should be expected to recall the above terms without the aid of a reference.

STANDARD OF PERFORMANCE

The above tasks should be performed to the following standards:

a) The graphics program should incorporate high-resolution graphics commands and those other Applesoft commands listed above or covered in COG-004, COG-005 or COG-006 activities. It should run without error and draw the intended image.

b) The student should be able to supply a correct definition for all of the terms. Definitions should be considered correct if they conform to definitions found in the current computer literature.

Selected Instructional Materials: An Apple II+, IIe, IIc, or IIgs computer with printer, monitor (preferably with color), and disk drive; an Applesoft reference manual; an Apple II utilities diskette or system master diskette; a formatted diskette; the *Computer Graphics Curriculum Project Master Diskette*; a piece of graph paper representing the Apple high-resolution graphics screen; and a pencil.

Teacher References

Recommended

Books:

Apple Computer, Inc. (1983). *Applesoft BASIC programmer's reference manual*. Reading, MA: Addison-Wesley.

Apple Computer, Inc. (1986). *A touch of Applesoft BASIC*. Cupertino, CA: Author.

Myers, R.E. (1984). *Microcomputer graphics for the Apple computer*. Reading, MA: Addison-Wesley.

Software:

Apple IIgs System Disk, ProDOS User's Disk, or *Apple II System Master (DOS 3.3)*.

Suggested

Book:

Bitter, G., & Severe, S. (1986). *Programming in BASIC for the Apple II series*. Reading, MA: Addison-Wesley.

Teacher Activities

1. Familiarize yourself with the Applesoft BASIC high-resolution commands HGR, HCOLOR, and HPLOT, and with the PEEK, POKE, and CALL commands.

2. Explain to the student the meaning of the terms frame buffer, pixel, display buffer, bit map, and high-resolution graphics display buffer (this last term is an Apple term). Demonstrate to the student how you can POKE a value into a frame buffer location and cause pixels to be lighted on the screen (see Chapter Two, *Microcomputer Graphics for the Apple Computer* by Myers). Also, demonstrate how the high-resolution aspect ratio of a display device can be measured and computed using the computer as a calculator.

3. Present the Applesoft commands HGR, HCOLOR, HPLOT, PEEK, POKE, and CALL to the student, preferably using the demonstration method or a self instructional module. The following are some reminders:

a. When explaining the HPLOT command, be sure to present it in all of its different forms. Describe the high-resolution screen.

b. Explain both the HGR and HGR2 commands.

c. Emphasize that HCOLORs 0-3 should not be used in the same program with HCOLORs 4-7.

4. Run and study all of the programs found in the student activity. Be sure you understand the logic of the programs and all of the commands used in them. If time permits, work through the student activities.

STUDENT REFERENCES

Recommended

Books:

Apple Computer, Inc. (1983). *Applesoft BASIC programmer's reference manual*. Reading, MA: Addison-Wesley.

Apple Computer, Inc. (1986). *A touch of Applesoft BASIC*. Cupertino, CA: Author.

Software:

Apple IIgs System Disk, ProDOS User's Disk, or *Apple II System Master (DOS 3.3).*

Suggested

Book:

Bitter, G., & Severe, S. (1986). *Programming in BASIC for the Apple II series*. Reading, MA: Addison-Wesley.

Myers, R.E. (1984). *Microcomputer graphics for the Apple computer*. Reading, MA: Addison-Wesley.

Student Activities:

1. Read about Apple high-resolution graphics and the Applesoft BASIC commands HGR, HCOLOR, and HPLOT in one of the references listed above. Then, key the following program into your computer:

[COG.007.1]

```
10 REM ** ACTIVITY SEVEN **
20 REM * COMPUTER GRAPHICS *
30 HGR
```

```
40 HCOLOR = 3
50 HPLOT 140,50
60 HPLOT TO 190,100
70 HPLOT 90,100 TO 140,50
80 END
```

RUN, debug, and SAVE this program with a unique name (COG.007.1).

To see another variation of the HPLOT command and to change the angle into a triangle, revise line 70 to read:

[COG.007.2]

```
70 HPLOT TO 90,100 TO 140,50
```

Try yet another version by DELeting lines 50,60, and 70 and adding:

[COG.007.3]

```
50 HPLOT 140,50 TO 190,100 TO 90,100 TO 140,50
```

Read about the POKE and PEEK commands in one of the above references and then add the following to have full screen graphics:

[COG.007.4]

```
35 POKE -16302,0
```

You may not notice much difference until you try to enter new commands from the keyboard and find that they no longer appear on the screen.

You can switch back from full page graphics to mixed text and graphics by a poke to -16303,0 or by using the HGR command again.

Delete line 35 and then change line 30 to read as follows.

[COG.007.5]

```
30 HGR2
```

It appears that the last two changes had the same result. However, they are different. Do you know how they differ?

If you have a color display device (monitor), make the following changes. These changes will allow you to see the different colors available in Apple high-resolution graphics. Even if you have a monochrome monitor, you can make these changes. The results will be different patterns of the same color.

add:

```
25 FOR C = 0 TO 7
60 FOR PAUSE = 0 TO 5000:NEXT PAUSE
70 NEXT C
75 TEXT:HOME
```

change:

```
30 HGR2
40 HCOLOR = 3
```

to:

```
30 HGR
40 HCOLOR = C
```

When finished, this is how your program should look:

[COG.007.6]

```
10 REM **  ACTIVITY  SEVEN  **
20 REM * COMPUTER GRAPHICS *
25 FOR C = 0 TO 7
30 HGR
40 HCOLOR = C
50 HPLOT 140,50 TO 190,100 TO 90,100 TO 140,50
60 FOR PAUSE = 0 TO 5000:NEXT PAUSE
70 NEXT C
75 TEXT:HOME
80 END
```

RUN, debug, and SAVE this program with a unique name (COG.007.6).

When you RUN it, you will see all of the eight high-resolution colors except the two blacks (black colors on a black background). If you are the type of person that needs to see things before you are really convinced, try adding the following two lines (the first line contains three commands) to the program:

[COG.007.7]

```
35 HCOLOR = 3:HPLOT 1,1:CALL 62454
37 IF C = 3 OR C = 7 THEN HGR
```

Now, try running the program. Are you convinced?

Line 35 contains a three command sequence that paints the screen. In this case, it is painted with the color number 3 (white). It can be painted with a different color by changing the HCOLOR command in line 35. The CALL 62454 calls a machine language subroutine found in ROM memory location # 62454 that paints the screen with the color used in the last plot. This is the reason for preceding the CALL command with the HCOLOR and the HPLOT

commands in line 35. Line 37 recalls the HGR command which changes the background back to black if either of the two whites are being used to draw the triangle. Take this opportunity to read about the CALL command in one of the above references.

2. Key the following program into your computer. It is very much like the one you were using in Part 1, above. You could just make some changes in the old program.

[COG.007.8]

```
10 REM **  ACTIVITY  SEVEN  **
20 REM * COMPUTER GRAPHICS *
30 HGR
40 HCOLOR = 3
50 HPLOT 190,50 TO 140,50
60 HPLOT TO 140,100 TO 190,100 TO 190,50
70 END
```

This program will draw a 50 pixel by 50 pixel square. However, since high-resolution pixels are not square just as low-resolution pixels are not, the block is not square. You will need to add a correction factor based on the aspect ratio of your screen similar to the correction factor you used when working with low-resolution graphics. **You will not use the same correction factor. You will need to compute it as described below.**

With the block on the screen, take a ruler and measure the height and width. Then use the following formula to compute the aspect ratio of your block.

ASPECT RATIO1 = HEIGHT/WIDTH

Now, make the following changes to use this correction factor in your program:

[COG.007.9]

add:

```
45 LET CY = .917 (use the correction factor you computed)
```

change:

```
50 HPLOT 190,50 TO 140,50
60 HPLOT TO 140,100 TO 190,100 TO 190,50
```

to:

```
50 HPLOT 190,50 * CY TO 140,50 * CY
60 HPLOT TO 140,100 * CY TO 190,100 * CY TO 190,50 * CY
```

You can also apply the correction factor to the X coordinates by following this procedure:

ASPECT RATIO2 = WIDTH/HEIGHT

[COG.007.10]

add:

```
45 CX = 1.09 (use the correction factor you computed with the above formula)
```

change:

```
50 HPLOT 190,50 TO 140,50
60 HPLOT TO 140,100 TO 190,100 TO 190,50
```

to:

```
50 HPLOT 190 * CX,50 TO 140 * CX, 50
60 HPLOT TO 140 * CX, 100 TO 190 * CX, 100 TO 190 * CX, 50
```

Although the squares will not be the same size, both examples should result in a square (or a close approximation).

3. If you have been using a color monitor, you have noticed that although you have assigned the color to be white (HCOLOR = 3) the vertical lines are violet or they may each be a different color, but not white. This is not a problem with your program, but is the result of the peculiar way that Applesoft changes colors on your monitor.

   If you are using a color monitor, try changing the line at 40 in COG-007.9 to use color 7 (the other white). **Be sure you make this change to COG.007.9, not COG.007.10.**

   change:

   ```
   40 HCOLOR = 3
   ```

   to:

   ```
   40 HCOLOR = 7
   ```

   This is the way your program should look after these changes:

   [COG.007.11]

   ```
   10 REM ** ACTIVITY SEVEN **
   20 REM * COMPUTER GRAPHICS *
   30 HGR
   40 HCOLOR = 7
   45 LET CY = .917
   50 HPLOT 190,50 * CY TO 140, 50 * CY
   60 HPLOT TO 140, 100 * CY TO 190, 100 * CY TO 190, 50 * CY
   70 END
   ```

   Now, when you RUN the program you will have blue vertical lines.

   To further illustrate this peculiarity, make the following changes:

   [COG.007.12]

   change:

   ```
   50 HPLOT 190,50 * CY TO 140, 50 * CY
   60 HPLOT TO 140, 100 * CY TO 190, 100 * CY TO 190, 50 * CY
   ```

to:

```
50 HPLOT 189,50 * CY TO 139,50 * CY
60 HPLOT TO 139,100 * CY TO 189,100 * CY TO 189,50 * CY
```

Now, when you RUN the program using the two whites (HCOLOR = 3 or HCOLOR = 7), you have the same problem. But, the vertical lines are drawn in green and orange instead of violet and blue. All of this is the result of Apple trying to save RAM memory which was much more expensive when Apples were first introduced.

To understand this, read about and have your instructor explain the terms PIXEL and FRAME BUFFER (sometimes called other things, such as, HIGH-RESOLUTION GRAPHICS DISPLAY BUFFER [Apple] or BIT MAP) if you do not already know their meaning.

The following is a brief explanation of what causes the above peculiarity: When a bit in the Apple frame buffer is set to a "1", the corresponding pixel on the display device screen is lighted (HCOLOR = 1,2,3,5,6, or 7). If the pixel being addressed has an odd number address (1,3,189,231, etc.) and neither the pixel on the right nor left of it is turned on, the color is either green or orange. If the pixel has an even number address (2,4,190,232, etc.) and neither the pixel on the right nor left of it is turned on, the color is either violet or blue. The color white is displayed only if two pixels which are side by side are set (turned on) at the same time. The Apple computer uses either color palette one (HCOLOR = 0,1,2,or 3) or color palette two (HCOLOR = 4,5,6, or 7) depending on the status of the high order bit of the byte in which the frame buffer bit which is presently being used is located. This arrangement allowed Applesoft to use eight colors (actually just six) while only devoting eight bits of frame buffer memory to describe seven pixels. Normally, using eight colors requires three bits per pixel. If you would like to know more about this read pages 22 - 25 in the *Apple IIe Technical Reference Manual*.

The only way to ensure that your vertical lines will be white is to draw two vertical lines side by side. Remember, white is displayed on your monitor only when two pixels which are side by side are lighted at the same time. This has been done in the following program. Key it in or change the previous program. Be sure the value in line 50 corresponds to the correction factor you computed for your display device.

[COG.007.13]

```
10 REM **  ACTIVITY  SEVEN  **
20 REM * COMPUTER GRAPHICS *
30 HGR
40 HCOLOR = 3
50 LET CY = .917
60 HPLOT 189,50 * CY TO 139,50 * CY TO 139,100 * CY
70 HPLOT TO 189,100 * CY TO 189,50 * CY
80 REM ***************************
90 REM *DRAWS THE XTRA LINES*
95 REM ***************************
100 HPLOT TO 140,50 * CY TO 140,100 * CY
110 HPLOT TO 190,50 * CY TO 190,100 * CY
120 END
```

4. To illustrate another problem caused by the way Apple uses colors, try changing the preceding program to the following (again, it helps if you have a color monitor, but it is not absolutely necessary):

[COG.007.14]

```
10 REM **  ACTIVITY SEVEN  **
20 REM * COMPUTER GRAPHICS *
30 HGR
40 HCOLOR = 1
80 LET CY = .917
90 HPLOT 190,50 * CY TO 140,50 * CY
100 HPLOT TO 140,100 * CY TO 190,100 * CY TO 190,50 * CY
110 END
```

Now, the program does not draw the vertical lines at all. The reason is that we are attempting to use the color green (HCOLOR = 1) and draw a vertical line on columns of pixels with even number addresses--190 and 140. Remember from the explanation above, only violet and blue can be used in a pixel with an even number address. Odd number addresses can use green and orange.

[COG.007.14.1]

change:

```
40 HCOLOR = 1
```

to:

```
40 HCOLOR = 2
```

Now with the color assigned to violet, the vertical lines will be drawn. Try the same thing with the colors 5 & 6 (orange and blue) as you did with 1 & 2 (green and violet). You will see similar results.

You will also encounter problems when you attempt to mix colors from the two palettes (mix HCOLOR = 1,2, or 3 with HCOLOR = 5,6, or 7). This is illustrated with the following programs. Key them into your computer and RUN them.

[COG.007.15]

```
10 REM **  ACTIVITY SEVEN  **
20 REM * COMPUTER GRAPHICS *
30 HGR
40 REM ************************
50 REM *  SCREEN  PAINTING   *
60 REM *  ROUTINE  (GREEN)   *
70 REM ************************
80 HCOLOR = 1:HPLOT 1,1:CALL 62454
90 HCOLOR = 5
100 REM ************************
110 REM * DRAWS  A  TRIANGLE  *
120 REM * IN ORANGE/HCOLOR=5*
130 REM ************************
140 HPLOT 140,50 TO 190,100 TO 90,100 TO 140,50
150 END
```

[COG.007.16]

Be sure to use YOUR correction factor in line 140 below.

```
10 REM **  ACTIVITY  SEVEN   **
20 REM * COMPUTER GRAPHICS *
30 HGR
40 REM ************************
50 REM *  SCREEN  PAINTING    *
60 REM *  ROUTINE  (VIOLET)   *
70 REM ************************
80 HCOLOR = 2:HPLOT 1,1:CALL 62454
90 HCOLOR = 6
100 REM ************************
110 REM *  DRAWS  A  SQUARE    *
120 REM *  IN  BLUE/HCOLOR=6   *
130 REM ************************
140 LET CY = .917
150 HPLOT 190,50 * CY TO 140,50 * CY
160 HPLOT TO 140,100 * CY TO 190,100 * CY TO 190,50 * CY
170 END
```

This problem is caused by the need for all bits within a byte describing a portion of the screen (seven adjacent pixels) to be within the same color palette. Therefore, when you change one pixel to a different palette, the Apple system automatically changes the other six adjacent horizontal pixels.

You will be happy to know that the color problems described in the last two parts of this activity are unique to the Apple II family of computers. You may encounter unique problems with other computers but not these.

5. As you have probably realized by now, microcomputers are well suited to drawing individual dots (turning on pixels). In this respect they are very much like the large programmable signs you see at the ball park or in front of banks. Other types of computer displays do a better job of drawing straight lines, but none can actually draw a curved line. As is done in the following program, computer graphics programs draw a series of short lines to simulate a curve. Key this program in, debug, and RUN it:

[COG.007.17]

Be sure to use YOUR correction factor in line 45 below.

```
10 REM **  ACTIVITY  SEVEN   **
20 REM * COMPUTER GRAPHICS *
30 HGR
40 HCOLOR = 3
45 LET CY = .917
50 REM ************************
60 REM * DRAWS  A  CIRCLE      *
70 REM * USING TRIGONOMETRY *
80 REM * FUNCTIONS            *
90 REM ************************
```

```
100 REM ************************
110 REM * ASSIGN  VALUES  FOR *
120 REM * CIRCLE  CENTER  AND *
130 REM * RADIUS              *
140 REM ************************
150 LET XCEN = 140
160 LET YCEN = 80
170 LET RAD = 30
180 REM ************************
190 REM * COMPUTE THE ANGLE *
200 REM * IN  RADIANS  FOR     *
210 REM * 48  CHORDS           *
220 REM ************************
230 LET ANG = 2 * 3.1416 / 48
240 LET BANG = ANG
250 REM ************************
260 REM * DRAW  THE  CIRCLE    *
270 REM ************************
280 HPLOT XCEN + RAD, YCEN
290 FOR N = 1 TO 48
300 LET X = XCEN + RAD * COS (ANG)
310 LET Y = YCEN + RAD * SIN (ANG) * CY
320 HPLOT TO X,Y
330 LET ANG = ANG + BANG
340 NEXT N
350 END
```

This program does not draw a very pleasing curve. However, it isn't the fault of the program. Despite the name, Applesoft high-resolution graphics just does not have sufficient resolution to draw a pleasing curve.

Have your instructor explain the logic of this program.

Even after an excellent explanation, you may not understand it. If so, do not be concerned. Just save the program and use it whenever you wish to draw circles. You can draw any size circle by changing the values assigned in lines 150, 160, and 170. However, if you are interested in mathematics and want to know more about the program and other programs for drawing circles, read about them in Chapter Five of *Microcomputer Graphics for the Apple Computer* by Myers.

Try changing the values in lines 150, 160, and 170 to draw different size circles at different locations. Do not be surprised if you get a fatal error when you draw a circle that is so large or so close to the edge of the screen that part of it falls off the screen. You will be shown how to guard against that possibility in the next activity.

6. When Applesoft graphic commands are combined with mathematics, the possibilities of drawing unusual images are unlimited. To illustrate, try running the following programs, written originally by Roy E. Myers. They can be found on the *Computer Graphics Curriculum Project Master Diskette.*

   With the above diskette in your disk drive, key in the following:

   RUN BOUNCING BALL

To exit this program, you must use CTRL C. Type TEXT to return to the text screen.

Now, try:

RUN MOIRE

You can press ESC to exit this program.

Next, try:

RUN RING

Type TEXT to return to the text screen.

7. Using the Applesoft commands covered in this activity and previous activities, write a high-resolution graphics program that will draw the two view drawing shown below. You may find the Apple high-resolution graphics screen graph paper in Appendix A to be helpful.

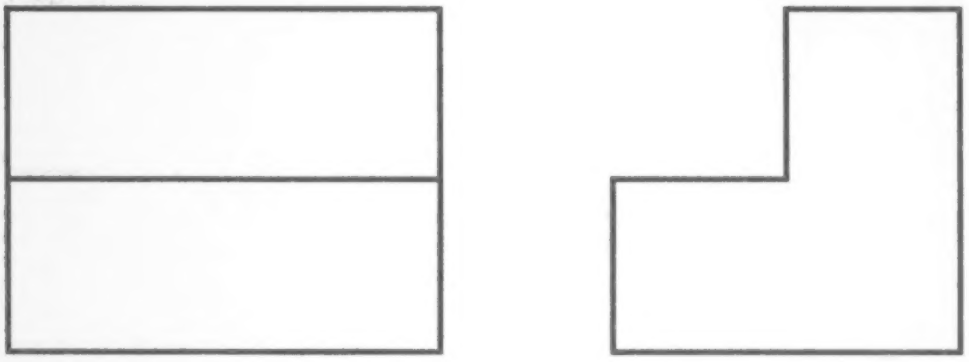


Figure 7--Two View Drawing

8. As an optional activity, select another image and write a program that will draw it. Try to choose something that uses solid lines and doesn't contain curve lines (circles are O.K.).

Evaluation Criteria: To evaluate the student's success in completing this activity, the teacher should give the student a two part test. The first part should be a performance test where the student is required to write a computer graphics program incorporating the Applesoft high-resolution commands HGR, HCOLOR, and HPLOT. The student should be instructed to use only those commands covered in this activity or those covered in activities COG-004, COG-005, and COG-006. The second part should be a recall test requiring the student to define the terms which were covered in this activity.

To be considered successful, the student's program should run on the computer without error and draw the intended image. On the second part, the student should be able to provide an acceptable definition for all of the terms. A term will be considered acceptable if it agrees with definitions found in current computer literature.

Computer Graphics Programming on an Apple II

GRAPHICS PROGRAMMING IN BASIC USING ARRAYS AND SUBPROGRAMS

Introduction: This activity is the fifth in a series of activities that is designed to teach the student how to draw computer images by programming an Apple computer with the BASIC language. This activity can be done on any Apple computer that uses Applesoft BASIC, the II+, IIe, IIc, or IIgs.

Enabling Objective: Before beginning this activity, the student should have completed Activity COG-007, Graphics Programming in BASIC Using Applesoft High-Resolution Commands. He or she should have demonstrated mastery of the objectives found in that activity before beginning this one.

Performance Objective (6-12 days):

TASK/COMPETENCY--COG-008

The student should be able to write a high-resolution graphics program using arrays, functions, and the BASIC commands GOSUB...RETURN, DIM and DEF FN. He or she should also be able to define the following terms associated with graphics programming:

clipping
window
array
matrix
point of intersection
end point
graphic transformations
algorithm

CONDITIONS

The above tasks should be performed under the following conditions:

a) The student should be provided an Apple computer with a disk drive, monitor, and a printer and an Applesoft BASIC reference manual when writing the graphics program.

b) The student should be expected to recall the definitions of the above terms without the aid of a reference.

STANDARD OF PERFORMANCE

The above tasks should be performed to the following standards:

a) The graphics program should incorporate high-resolution graphics commands and those other Applesoft commands listed above or covered in COG-004, COG-005, COG-006, or COG-007 activities. It should run without error and draw the intended image.

b) The student should be able to supply a correct definition for all of the terms. Definitions should be considered correct if they conform to definitions found in the current computer literature.

Selected Instructional Materials: An Apple II+, IIe, IIc, or IIgs computer with printer, monitor (preferably with color), and disk drive; an Applesoft reference manual; an Apple II utilities diskette or system master diskette; a formatted diskette; the *Computer Graphics Curriculum Project Master Diskette*; a piece of graph paper representing the Apple high-resolution graphics screen; and a pencil.

Teacher References

Recommended

Books:

Apple Computer, Inc. (1983). *Applesoft BASIC programmer's reference manual*. Reading, MA: Addison-Wesley.

Apple Computer, Inc. (1986). *A touch of Applesoft BASIC*. Cupertino, CA: Author.

Myers, R.E. (1984). *Microcomputer graphics for the Apple computer*. Reading, MA: Addison-Wesley.

Software:

Apple IIgs System Disk, ProDOS User's Disk, or *Apple II System Master (DOS 3.3)*.

Suggested

Book:

Bitter, G., & Severe, S. (1986). *Programming in BASIC for the Apple II series*. Reading, MA: Addison-Wesley.

Teacher Activities

1. Familiarize yourself with the Applesoft BASIC commands GOSUB...RETURN, DIM and DEF FN. Also familiarize yourself with the use of arrays and with the library functions available with Applesoft.

2. Look up the meaning of the terms clipping, window, array, matrix, point of intersection,

graphic transformations, and end point; review the student activities to see how these terms fit into the activities; and then explain them to the students.

3. Present the Applesoft commands GOSUB...RETURN, DIM, and DEF FN to the student, preferably using the demonstration method or a self instructional module. The following are some reminders:

   a. When explaining the GOSUB...RETURN commands, be sure to emphasize that an END statement must precede a subroutine. If it doesn't the subroutine may be executed when it isn't intended.

   b. Also explain that good programming techniques place the subroutine at the end of a program listing (the subroutine has higher line numbers than the main). This makes it easier for someone to read and understand the program. However, they may encounter many programs written in Applesoft where the subroutines are found at the beginning of the program listing. The reason they are found at the beginning is because Applesoft programs will execute faster with them in this position. However, many times it makes the program impossible to understand by someone other than the original programmer. Therefore, it is considered to be a poor programming technique. Top down programming is preferred.

   c. Be extra deliberate when explaining the commands DEF FN and FN. For some reason, students find these commands very abstract and difficult to understand. It may be best to use very simple examples to begin with.

   d. When explaining arrays explain that they are analogous to a set of file drawers. If your students are not mathematically inclined, it may be better than explaining them as the computer's version of subscripted numbers.

4. Run and study all of the programs found in the student activity. Be sure you understand the logic of the programs and all of the commands used in them. If time permits, work through the student activities.

5. If you choose to have your students do the optional activity in part 6, it is best if they design the clipping routine for a drawing with only horizontal or vertical lines. A clipping routine for inclined lines should only be attempted by students who understand the mathematical formula for a straight line.

Student References

Recommended

Books:

Apple Computer, Inc. (1983). *Applesoft BASIC programmer's reference manual*. Reading, MA: Addison-Wesley.

Apple Computer, Inc. (1986). *A touch of Applesoft BASIC*. Cupertino, CA: Author.

Software:

Apple IIgs System Disk, ProDOS User's Disk, or *Apple II System Master (DOS 3.3)*.

Suggested

Book:

Bitter, G., & Severe, S. (1986). *Programming in BASIC for the Apple II series*. Reading, MA: Addison-Wesley.

Myers, R.E. (1984). *Microcomputer graphics for the Apple computer*. Reading, MA: Addison-Wesley.

Student Activities:

1. Read about Applesoft BASIC commands GOSUB and RETURN in one of the references listed above. Then, LOAD the circle program [COG.007.17] from the last activity, COG-007.

   old program from Activity COG-007:

[COG.007.17]

```
10 REM **  ACTIVITY SEVEN  **
20 REM * COMPUTER GRAPHICS *
30 HGR
40 HCOLOR = 3
45 LET CY = .917
50 REM ************************
60 REM * DRAWS  A CIRCLE      *
70 REM * USING TRIGONOMETRY *
80 REM * FUNCTIONS           *
90 REM ************************
100 REM ***********************
110 REM * ASSIGN  VALUES  FOR *
120 REM * CIRCLE  CENTER  AND *
130 REM * RADIUS              *
140 REM ***********************
150 LET XCEN = 140
160 LET YCEN = 80
170 LET RAD = 30
180 REM ***********************
190 REM * COMPUTE THE  ANGLE *
200 REM * IN  RADIANS  FOR    *
210 REM * 48  CHORDS          *
220 REM ***********************
230 LET ANG = 2 * 3.1416 / 48
240 LET BANG = ANG
250 REM ***********************
```

```
260 REM * DRAW THE CIRCLE   *
270 REM ************************
280 HPLOT XCEN + RAD, YCEN
290 FOR N = 1 TO 48
300 LET X = XCEN + RAD * COS (ANG)
310 LET Y = YCEN + RAD * SIN (ANG) * CY
320 HPLOT TO X,Y
330 LET ANG = ANG + BANG
340 NEXT N
350 END
```

Change it into a subroutine. DELete lines 100 through 170, add a RETURN statement, and renumber the old program. You no longer need lines 100 through 170 because you will depend on the main program to assign values to the circle center and the circle radius. Higher line numbers will make it easier to use the subroutine with other programs. Review and use the advanced editing techniques with the ESC...arrow keys or ESC...I, J, K, and M.

This subroutine will not work by itself. You must supply values for the variables by calling the subroutine from a main program which supplies them or by assigning values in immediate mode.

The following is the above program transformed into a subroutine:

[COG.008.1]

```
1000 REM * CIRCLE SUBROUTINE *
1010 REM ************************
1020 REM ** ACTIVITY EIGHT  **
1030 REM *COMPUTER GRAPHICS*
1040 HGR
1050 HCOLOR = 3
1060 LET CY = .917
1070 REM ************************
1080 REM *DRAWS A CIRCLE    *
1090 REM *USING TRIGONOMETRY*
1100 REM *FUNCTIONS         *
1110 REM ************************
1120 REM ************************
1130 REM * COMPUTE THE ANGLE*
1140 REM * IN RADIANS FOR   *
1150 REM * 48 CHORDS       *
1160 REM ************************
1170 LET ANG = 2 * 3.1416 / 48
1180 LET BANG = ANG
1190 REM ************************
1200 REM * DRAW THE CIRCLE  *
```

```
1210 REM ************************
1220 HPLOT XCEN + RAD, YCEN
1230 FOR N = 1 TO 48
1240 LET X = XCEN + RAD * COS (ANG)
1250 LET Y = YCEN + RAD * SIN (ANG) * CY
1260 HPLOT TO X,Y
1270 LET ANG = ANG + BANG
1280 NEXT N
1290 RETURN
1300 END
```

SAVE this subroutine in a separate file so that you can use it with other programs. After it has been saved and with it still in the computer's memory, add lines 10 through 140 as shown below. This part (lines 10-140) is known as the main program. It is used to call the subroutine:

[COG.008.1.1]

```
10 REM **    ACTIVITY TEN   **
20 REM * COMPUTER GRAPHICS *
30 REM ************************
40 REM * ASSIGN   VALUES     *
50 REM * BEFORE CALLING SUB*
60 REM ************************
70 LET XC = 100
80 LET YC = 100
90 LET RAD = 40
100 REM ************************
110 REM * CALL CIRCLE SUB    *
120 REM ************************
130 GOSUB 1000
140 END


1000 REM *CIRCLE  SUBROUTINE*
1010 REM ************************
1020 REM ** ACTIVITY EIGHT   **
1030 REM *COMPUTER GRAPHICS*
1040 HGR
1050 HCOLOR = 3
1060 LET CY = .917
1070 REM ************************
1080 REM *DRAWS  A  CIRCLE     *
1090 REM *USING TRIGONOMETRY*
1100 REM *FUNCTIONS            *
1110 REM ************************
1120 REM ************************
1130 REM * COMPUTE THE ANGLE*
1140 REM * IN RADIANS  FOR    *
1150 REM * 48 CHORDS          *
1160 REM ************************
1170 LET ANG = 2 * 3.1416 / 48
1180 LET BANG = ANG
1190 REM ************************
1200 REM * DRAW  THE  CIRCLE  *
```

```
1210 REM ************************
1220 HPLOT XCEN + RAD, YCEN
1230 FOR N = 1 TO 48
1240 LET X = XCEN + RAD * COS (ANG)
1250 LET Y = YCEN + RAD * SIN (ANG) * CY
1260 HPLOT TO X,Y
1270 LET ANG = ANG + BANG
1280 NEXT N
1290 RETURN
1300 END
```

If your main program is to be used to call other drawing subroutines or this subroutine more than once, you should move additional statements to the main. Move the HGR and the HCOLOR statements to the main with the following changes:

change:

[COG.008.1.2]

```
70 LET XC = 100
80 LET YC = 100
90 LET RAD = 40
100 REM **********************
110 REM * CALL CIRCLE SUB  *
120 REM **********************
130 GOSUB 1000
140 END
```

to:

```
70 HGR
80 HCOLOR = 3
90 LET XC = 120
100 LET YC = 80
110 LET RAD = 30
120 REM **********************
130 REM * CALL CIRCLE SUB  *
140 REM **********************
150 GOSUB 1000
160 END
```

Now, DELete lines 1040 and 1050 from the subroutine.

RUN, debug, and save your program with a unique name (COG.008.1.2). You can RUN the program again and again.

The preceding changes will allow you to call the subroutine a number of times without erasing the screen with each call. To take advantage of this feature, make the following changes:

add:

```
75 FOR XC = 35 TO 235 STEP 50
155 NEXT XC
```

delete:

```
90 XC = 120
```

when finished, your program should LIST like so:

[COG.008.1.3]

```
10 REM ** ACTIVITY  EIGHT  **
20 REM * COMPUTER GRAPHICS *
30 REM ************************
40 REM * ASSIGN  VALUES      *
50 REM * BEFORE CALLING SUB*
60 REM ************************
70 HGR
75 FOR XC = 35 TO 235 STEP 50
80 HCOLOR = 3
100 LET YC = 80
110 LET RAD = 30
120 REM ************************
130 REM * CALL  CIRCLE  SUB   *
140 REM ************************
150 GOSUB 1000
155 NEXT XC
160 END
1000 REM *CIRCLE  SUBROUTINE*
1010 REM ************************
1020 REM **  ACTIVITY  EIGHT  **
1030 REM *COMPUTER  GRAPHICS*
1060 LET CY = .917
1070 REM ************************
1080 REM *DRAWS  A  CIRCLE     *
1090 REM *USING TRIGONOMETRY*
1100 REM *FUNCTIONS            *
1110 REM ************************
1120 REM ************************
1130 REM * COMPUTE THE  ANGLE*
1140 REM * IN  RADIANS  FOR    *
1150 REM * 48  CHORDS          *
1160 REM ************************
1170 LET ANG = 2 * 3.1416 / 48
1180 LET BANG = ANG
1190 REM ************************
1200 REM * DRAW  THE  CIRCLE   *
1210 REM ************************
1220 HPLOT XCEN + RAD, YCEN
```

87

```
1230 FOR N = 1 TO 48
1240 LET X = XCEN + RAD * COS (ANG)
1250 LET Y = YCEN + RAD * SIN (ANG) * CY
1260 HPLOT TO X,Y
1270 LET ANG = ANG + BANG
1280 NEXT N
1290 RETURN
1300 END
```

RUN, debug, and SAVE this program with a unique name (COG.008.1.3).

In addition to describing the variables XC, YC, and RAD in the main program, you can elect to describe others. The following changes will allow you to use this subroutine to draw polygons as well as simulated circles.

[COG.008.1.4]

add:

```
90 LET CHRD = 6
```

change:

```
1170 LET ANG = 2 * 3.1416 / 48
1230 FOR N = 1 TO 48
```

to:

```
1170 LET ANG = 2 * 3.1416 / CHRD
1230 FOR N = 1 TO CHRD
```

RUN, debug, and SAVE this program with a unique name (COG.008.1.4).

Use it to draw different polygons by changing the value of CHRD.

As an optional activity, use the INPUT or GET statements to interact with the user and draw a polygon with any number of different sides.

2. Lines 1240 and 1250 in the circle subroutine use built-in functions supplied by Applesoft. Some of the demonstration programs that you have RUN in previous activities have also made use of built-in functions. Read about built-in Applesoft functions in one of the above references. Then, try the RND (random number generator) and the INT (integer) functions with the following program.

To exit this program, you must use CTRL C. Also, remember to use the TEXT command to exit the graphics mode.

[COG.008.2]

```
10 REM **   ACTIVITY   TEN   **
20 REM * COMPUTER GRAPHICS *
30 HGR
40 REM ***************************
50 REM * CREATE A  RANDOM    *
60 REM * COLOR               *
70 REM ***************************
80 LET C = INT (8 * RND(1))
90 HCOLOR = C
100 REM ***************************
110 REM *CREATE RANDOM COOR-*
120 REM *DINATES  FOR  ONE      *
130 REM *END  POINT  OF  LINE   *
140 REM ***************************
150 LET X = INT (279 * RND (1))
160 LET Y = INT (159 * RND (1))
170 REM ***************************
180 REM * PLOT  LINE  FROM      *
190 REM * SCREEN  CENTER  TO    *
200 REM * RANDOM  POINT        *
210 REM ***************************
220 HPLOT X,Y TO 140,80
230 GOTO 20
240 END
```

RUN, debug, and SAVE this program with a unique name (COG.008.2).

Line 80 uses the INT and RND functions to generate a random number (RND) between 0 and 8. Since the function RND (1) will generate random numbers between 0.0 and 1.0, the generated number is multiplied times 8 to make the range between 0.0 and 8.0. The HCOLOR command expects an integer. Therefore, the INT function is then used to make the generated number an integer. This function truncates the number (it eliminates all parts of the number to the right of the decimal point). This means that the number 8.9999 will become an 8, 7.0001 will become a 7, 1.4231 will become a 1, etc.

Similarly, the statements at lines 150 and 160 are used to generate a set of end point coordinates that are random integers with ranges, $0 < X < 280$ and $0 < Y < 160$.

3. The use of arrays in graphics programming is very common. Read about using arrays (matrices) and the DIM statement in one of the above references. Also, review the use of the statements READ, RESTORE, and DATA. Then, key the following program into your computer's memory:

When you RUN this program, be careful and only input values for SCALE and XLOC and YLOC that will fall within the screen window ($0 < X < 280$ and $0 < Y < 160$). For example, SCALE = 4.0, times 80 (largest X coordinate), plus an X LOCATION of 20 would equal 340. Since 340 is greater than 279, the largest permissible X coordinate, an error will occur.

[COG.008.3]

```
10 REM **  ACTIVITY  EIGHT  **
20 REM * COMPUTER GRAPHICS *
30 REM ************************
40 REM * DIMENSION  ARRAYS *
50 REM ************************
60 DIM X(9), Y(9)
70 REM  ************************
80 REM  * OBTAIN  INPUT  FOR  *
90 REM  * SCALE  &  LOCATION *
100 REM ************************
110 INPUT "WHAT SIZE?";SCALE
120 INPUT "WHAT LOCATION?";XLOC,YLOC
130 REM ************************
140 REM * READ  AND  TRANS-  *
150 REM * FORM  DATA         *
160 REM ************************
170 FOR N = 1 TO 9
180 READ X(N),Y(N)
190 LET X(N) = X(N) * SCALE + XLOC
200 LET Y(N) = Y(N) * SCALE + YLOC
210 NEXT N
220 REM ************************
230 REM * DRAW  FIGURE        *
240 REM ************************
250 HGR:HCOLOR = 3
260 HPLOT X(1),Y(1)
270 FOR N = 2 TO 9
280 HPLOT TO X(N),Y(N)
290 NEXT N
300 REM ************************
310 REM * DATA  STORED        *
320 REM * AS  X, Y  PAIRS     *
330 REM ************************
340 DATA 0,0, 0,40, 80,40
350 DATA 80,0, 50,0, 50,20
360 DATA 30,20, 30,0, 0,0
370 END
```

RUN, debug, and SAVE this program with a unique name (COG.008.3).

This program allows the user to draw the image any size and at any location on the screen. It does this by transforming, but not changing permanently, the original data. Thus, it can be said that the purpose of this program is to perform graphics transformations. The particular types of transformation used in this example are known as a translation and scaling. Translation has to do with moving the object right, left, up, or down on the screen. Scaling has to do with changing the size of the object to make it larger or smaller. Although it isn't done in this example, a scaling transformation can also stretch the object--change the width without changing the height or vice versa.

4. In addition to the built-in functions available to you, Applesoft BASIC allows you to define and use your own functions. In one of the above references, read how to do this with the DEF FN and FN statements.

   In the previous program, the familiar aspect ratio correction factor is missing. Change the program to add the correction factor (be sure to use the one you computed). You can use the DEF FN and FN statements to do this and to perform the transformations made by lines 190 and 200. Make the following changes to accomplish this:

   add:

```
121 REM ************************
122 REM * DEFINE  FUNCTION   *
123 REM ************************
124 DEF FN XC(X) = X * SCALE + XLOC
125 DEF FN YC(Y) = Y * SCALE * .917 + YLOC
```

   change:

```
190 LET X(N) = X(N) * SCALE + XLOC
200 LET Y(N) = Y(N) * SCALE + YLOC
```

   to:

```
190 LET X(N) = FN XC(X(N))
200 LET Y(N) = FN YC(Y(N))
```

   When changed, the program should look like this:

[COG.008.4]

```
10 REM **  ACTIVITY  EIGHT    **
20 REM * COMPUTER GRAPHICS *
30 REM ************************
40 REM * DIMENSION  ARRAYS  *
50 REM ************************
60 DIM X(9), Y(9)
70 REM  ************************
80 REM  * OBTAIN  INPUT  FOR   *
90 REM  * SCALE  &  LOCAT ION  *
100 REM ************************
110 INPUT "WHAT SIZE?";SCALE
120 INPUT "WHAT LOCATION?";XLOC,YLOC
121 REM ************************
122 REM * DEFINE  FUNCTION   *
123 REM ************************
124 DEF FN XC (X) = X * SCALE + XLOC
125 DEF FN YC (Y) = Y * SCALE * .917 + YLOC
130 REM ************************
140 REM * READ  AND  TRANS -   *
150 REM * FORM  DATA           *
160 REM ************************
170 FOR N = 1 TO 9
```

```
180 READ X(N),Y(N)
190 LET X(N) = FN XC(X(N))
200 LET Y(N) = FN YC(Y(N))
210 NEXT N
220 REM ************************
230 REM * DRAW  FIGURE         *
240 REM ************************
250 HGR : HCOLOR = 3
260 HPLOT X(1),Y(1)
270 FOR N = 2 TO 9
280 HPLOT TO X(N),Y(N)
290 NEXT N
300 REM ************************
310 REM * DATA  STORED         *
320 REM * AS  X, Y  PAIRS      *
330 REM ************************
340 DATA 0,0, 0,40, 80,40
350 DATA 80,0, 50,0, 50,20
360 DATA 30,20, 30,0, 0,0
370 END
```

RUN, debug, and SAVE this program using a unique name (COG.008.4).

5. The programs you have used in this activity all work well if the user is careful not to cause the program to draw the circle or other images outside of the screen window (none of the X values are less than 0 nor greater than 279 and none of the Y values are less than 0 nor greater than 159). However, this is not satisfactory for many applications because it is very easy to slip up and request dimensions for an image that cannot be drawn inside of the window. In this part of the activity, you will change the circle subroutine to prevent this from happening.

   LOAD the following program from Part 1 of this activity.

   [COG.008.1.2]

```
10 REM **    ACTIVITY  TEN    **
20 REM * COMPUTER GRAPHICS *
30 REM ************************
40 REM * ASSIGN  VALUES      *
50 REM * BEFORE  CALLING SUB*
60 REM ************************
70 HGR
80 HCOLOR = 3
90 LET XC = 120
100 LET YC = 80
110 LET RAD = 30
120 REM ************************
130 REM * CALL  CIRCLE  SUB   *
140 REM ************************
150 GOSUB 1000
160 END
```

```
1000 REM *CIRCLE  SUBROUTINE*
1010 REM ************************
1020 REM ** ACTIVITY  EIGHT   **
1030 REM *COMPUTER GRAPHICS*
1060 LET CY = .917
1070 REM ************************
1080 REM *DRAWS  A  CIRCLE      *
1090 REM *USING TRIGONOMETRY*
1100 REM *FUNCTIONS             *
1110 REM ************************
1120 REM ************************
1130 REM * COMPUTE THE  ANGLE*
1140 REM * IN  RADIANS  FOR     *
1150 REM * 48  CHORDS           *
1160 REM ************************
1170 LET ANG = 2 * 3.1416 / 48
1180 LET BANG = ANG
1190 REM ************************
1200 REM * DRAW  THE  CIRCLE   *
1210 REM ************************
1220 HPLOT XCEN + RAD, YCEN
1230 FOR N = 1 TO 48
1240 LET X = XCEN + RAD * COS (ANG)
1250 LET Y = YCEN + RAD * SIN (ANG) * CY
1260 HPLOT TO X,Y
1270 LET ANG = ANG + BANG
1280 NEXT N
1290 RETURN
1300 END
```

Make the following changes to prevent the individual line (chord) from being drawn if either of its end points is outside of the Apple window. This is done by computing all of the points used to draw the circle, storing them in an array, and then examining each pair of end points before the individual line (chord) is drawn.

add:

```
1040 DIM X(49),Y(49)
```

change:

```
1200 REM * DRAW  THE  CIRCLE   *
1210 REM ************************
1220 HPLOT XCEN + RAD, YCEN
1230 FOR N = 1 TO 48
1240 LET X = XCEN + RAD * COS (ANG)
1250 LET Y = YCEN + RAD * SIN (ANG) * CY
1260 HPLOT TO X,Y
1270 LET ANG = ANG + BANG
1280 NEXT N
1290 RETURN
1300 END
```

93

to:

```
1200 REM * COMPUTE THE COORD*
1210 REM * FOR THE CIRCLE &   *
1220 REM * STORE THEM IN AN   *
1230 REM * ARRAY              *
1240 REM ********************
1250 LET X(1) = XCEN + RAD: LET Y(1) = YCEN
1260 FOR N = 2 TO 49
1270 LET X(N) = XCEN + RAD * COS (ANG)
1280 LET Y(N) = YCEN + RAD * SIN (ANG) * CY
1290 LET ANG = ANG + BANG
1300 NEXT N
1310 REM ********************
1320 REM * CHECK EACH END PT*
1330 REM * OF LINE BEFORE    *
1340 REM * DRAWING--DON'T    *
1350 REM * DRAW IF EITHER    *
1360 REM * IS NOT IN WINDOW *
1370 REM ********************
1380 FOR N = 1 TO 48
1390 FOR M = N TO N + 1
1400 IF X(M) > 279 OR X(M) < 0 GOTO 1440
1410 IF Y(M) > 159 OR Y(M) < 0 GOTO 1440
1420 NEXT M
1430 HPLOT X(N),Y(N) TO X(N+1),Y(N+1)
1440 NEXT N
1450 RETURN
1460 END
```

Next, change the main program to allow the user to input the coordinates for the circle center:

change:

```
90 LET XC = 120
100 LET YC = 80
```

to:

```
90 PRINT "WHAT ARE THE X,Y COORDINATES"
100 INPUT "OF YOUR CIRCLE CENTER?";XCEN,YCEN
```

When all changes have been made, the completed program should like this:

[COG.008.5]

```
10 REM **    ACTIVITY   TEN    **
20 REM * COMPUTER GRAPHICS *
30 REM ************************
40 REM * ASSIGN  VALUES       *
50 REM * BEFORE  CALLING SUB*
60 REM ************************
70 HGR
80 HCOLOR = 3
90 PRINT "WHAT ARE THE X,Y COORDINATES"
100 INPUT "OF YOUR CIRCLE CENTER?";XCEN,YCEN
110 LET RAD = 30
120 REM ************************
130 REM * CALL  CIRCLE  SUB    *
140 REM ************************
150 GOSUB 1000
160 END

1000 REM *CIRCLE  SUBROUTINE*
1010 REM ************************
1020 REM ** ACTIVITY    EIGHT  **
1030 REM *COMPUTER GRAPHICS*
1040 DIM X(49),Y(49)
1060 LET CY = .917
1070 REM ************************
1080 REM *DRAWS  A  CIRCLE       *
1090 REM *USING TRIGONOMETRY*
1100 REM *FUNCTIONS              *
1110 REM ************************
1120 REM ************************
1130 REM * COMP'ITE  THE  ANGLE*
1140 REM * IN  RADIANS  FOR     *
1150 REM * 48  CHORDS           *
1160 REM ************************
1170 LET ANG = 2 * 3.1416 / 48
1180 LET BANG = ANG
1190 REM ************************
1200 REM * COMPUTE THE  COORD*
1210 REM * FOR  THE  CIRCLE  &   *
1220 REM * STORE  THEM  IN  AN   *
1230 REM * ARRAY                 *
1240 REM ************************
1250 LET X(1) = XCEN + RAD: LET Y(1) = YCEN
1260 FOR N = 2 TO 49
1270 LET X(N) = XCEN + RAD * COS (ANG)
1280 LET Y(N) = YCEN + RAD * SIN (ANG) * CY
1290 LET ANG = ANG + BANG
1300 NEXT N
1310 REM ************************
```

```
1320 REM * CHECK  EACH  END  PT*
1330 REM * OF  LINE  BEFORE     *
1340 REM * DRAWING- -DON'T      *
1350 REM * DRAW  IF  EITHER     *
1360 REM * IS  NOT  IN  WINDOW  *
1370 REM ************************
1380 FOR N = 1 TO 48
1390 FOR M = N TO N + 1
1400 IF X(M) > 279 OR X(M) < 0  GOTO 1440
1410 IF Y(M) > 159 OR Y(M) < 0  GOTO 1440
1420 NEXT M
1430 HPLOT X(N),Y(N) TO X(N+1),Y(N+1)
1440 NEXT N
1450 RETURN
1460 END
```

RUN, debug, and SAVE your program. When running it, be sure to test the clipping function. Try to draw circles close to the edge of the window.

6. The procedure (algorithm) at lines 1380-1440 works well for a circle because the lines that are being used to draw the circle are very short. However, if this program is used to draw a polygon with longer chords, the results will not be satisfactory. An example of this is illustrated in Figure 8 below. The only parts of the three polygons that would be drawn using the procedure used in the circle program are lines 1,2 and 9,10. When one of the end points is outside the window, the line is not drawn even if the other end point is in the middle of the screen. The logic of this procedure (algorithm) is as follows:

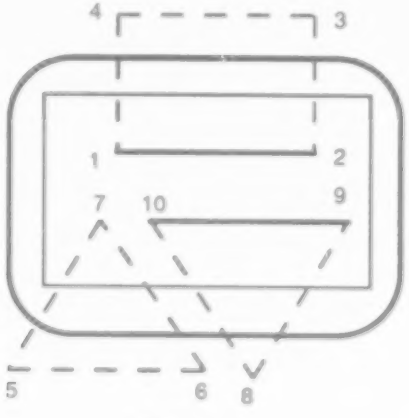


Figure 8.

a. Determine the coordinates of all the points in the polygon (simulated circle).
b. Examine the end points of each line in the polygon
    1) If either end point is outside the window ($0 < X < 279$ or $0 < Y < 159$) do not draw the line and proceed to examine the next two end points.
    2) Otherwise, draw the line.

A better but more complicated procedure is where a new end point is computed when one end point falls outside the window as in Figure 9. The procedure (algorithm) is changed to draw from the end point inside the window to a point on the edge of the window when only one end point is outside. This new point will be at the intersection of the original line and the edge of the window.

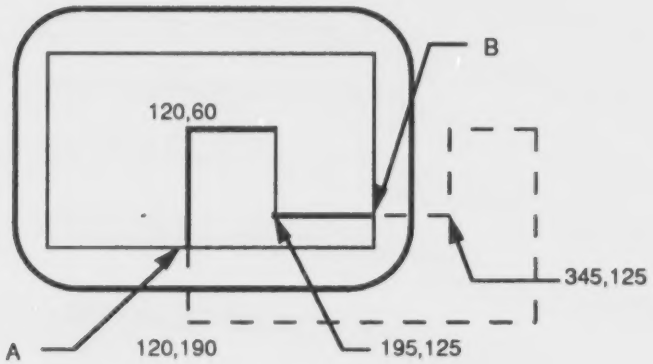


Figure 9.

Using this procedure (algorithm) on the image in Figure 9, new end points will be computed for the lines passing through points A and B. New coordinates of 120, 159 will be computed for point A, because the end point at 120, 190 has a Y coordinate larger than 159. New end point coordinates of 279, 125 are computed for point B, because the end point at 345, 125 has an X coordinate larger than 279.

Using this procedure to draw the rectangle, or any polygon with only horizontal or vertical lines, is not as complicated as it is for a polygon with inclined lines (see Figure 10). The logic that could be used in this special case (with only vertical or horizontal lines) is as follows:

a. Determine the coordinates of all the points in the polygon.
b. Examine the end points of each line in the polygon.
    1) If both end points are outside the window and the line does not pass through the window, do not draw the line and proceed to examine the next line.
    2) If both end points are outside the window and the line passes through the window, compute two new endpoints.
    2) Otherwise, examine each X and each Y coordinate as follows:
        (s) If an X coordinate is less than 0, then assign it a value of 0.
        (b) If an X coordinate is greater than 279, then assign it a value of 279.
        (c) If a Y coordinate is less than 0, then assign it a value of 0.
        (d) If a Y coordinate is greater than 159, then assign it a value of 159.
    3) Draw the line.

As an optional activity, change the program, [COG.008.4], in Part 4 to add a routine for clipping those lines that fall outside the window. The routine should work as described above.

The logic for a generalized procedure (algorithm) that could be used when drawing any polygon including the simulated circle and polygons with horizontal and vertical sides is too complicated for this course (see Figure 10). However, if you are an exceptional mathematics student and would like to learn more about it, you can read about clipping lines in Myers' *Microcomputer Graphics for the Apple Computer*.

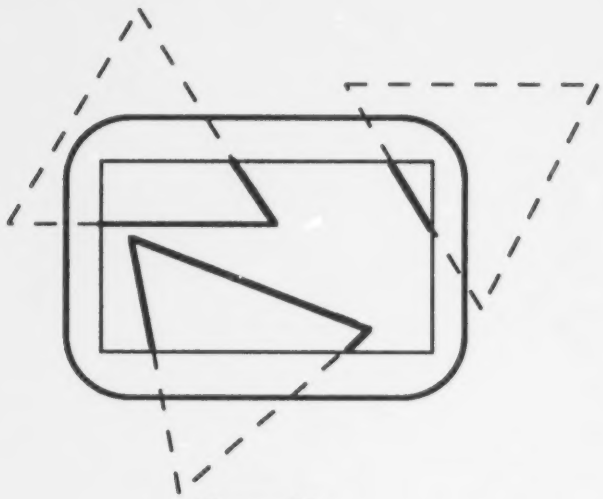


Figure 10.

Evaluation Criteria: To evaluate the student's success in completing this activity, the teacher should give the student a two part test. The first part should be a performance test where the student is required to write a high-resolution computer graphics program incorporating arrays, functions, and the BASIC commands DIM, GOSUB...RETURN and DEF FN. The student should be instructed to use only those commands covered in this activity or those covered in activities COG-004, COG-005, COG-006, and COG-007. The second part should be a recall test requiring the student to define the terms which were covered in this activity.

To be considered successful, the student's program should run on the computer without error and draw the intended image. On the second part, the student should be able to provide an acceptable definition for all of the terms. A term will be considered acceptable if it agrees with definitions found in current computer literature.

Computer Graphics Programming on an Apple II

GRAPHICS PROGRAMMING IN BASIC USING INPUT COMMANDS

Introduction: This activity is the sixth in a series of activities that is designed to teach the student how to draw computer images by programming an Apple computer with the BASIC language. This activity can be done on any Apple computer that uses Applesoft BASIC, the II+, IIe, IIc, or IIgs, and has an Apple mouse attached.

Enabling Objective: Before beginning this activity, the student should have completed Activity COG-008, Graphics Programming in BASIC Using Arrays and Subprograms. He or she should have demonstrated mastery of the objectives found in that activity before beginning this one.

Performance Objective (6-12 days):

TASK/COMPETENCY--COG-009

The student should be able to write an interactive graphics program which utilizes a graphics input device through the DOS and ProDOS commands IN# and PR#. He or she should also be able to define the following term relative to graphics programming:

input constraints

CONDITIONS

The above tasks should be performed under the following conditions:

a) The student should be provided an Apple computer with a disk drive, monitor, mouse (preferably an Apple mouse) and a printer and an Applesoft BASIC reference manual when writing the graphics program.

b) The student should be expected to recall the definition of the above term without the aid of a reference.

STANDARD OF PERFORMANCE

The above tasks should be performed to the following standards:

a) The graphics program should incorporate high-resolution graphics commands and those other Applesoft commands listed above or covered in COG-004, COG-005, COG-006, COG-007, and COG-008 activities. It should run without error and permit the user to input graphical data by using an Apple mouse or other input device.

b) The student should be able to supply a correct definition for the term that is similar to a definition found in the current computer literature.

Selected Instructional Materials: An Apple II+, IIe, IIc, or IIgs computer with printer, monitor (preferably with color), input device (preferably an Apple mouse), and disk drive; an Applesoft reference manual; an Apple II utilities diskette or system master diskette; a formatted diskette; the Computer Graphics Curriculum Project Master Diskette; a piece of graph paper representing the Apple high-resolution graphics screen; and a pencil.

Teacher References

Recommended

Books:

Apple Computer, Inc. (1983). *Applesoft BASIC programmer's reference manual.* Reading, MA: Addison-Wesley.

Apple Computer, Inc. (1983). *Apple II DOS user's manual.* Cupertino, CA: Author.

Apple Computer, Inc. (1987). *BASIC programming with ProDOS.* Reading, MA: Addison-Wesley.

Software:

Apple IIgs System Disk, ProDOS User's Disk, or *Apple II System Master (DOS 3.3).*

Teacher Activities

1. Familiarize yourself with the Applesoft BASIC library function CHR$ and the DOS and ProDOS commands IN# and PR#.

2. Look up the meaning of the term input constraint; review the student activities to see how it fits into the activities; and then explain it to the students. Be sure to explain the different types--angular, linear, grid, etc.

3. Demonstrate to the students how to use DOS or ProDOS commands in an Applesoft BASIC program. The following are some reminders:

   a. Before explaining the function CHR$, be sure the students understand ASCII codes. Show them an ASCII code table and explain how these codes (decimal) can be used with the CHR$ function.

   b. Even if you have covered it before, show them the slots on the Apple motherboard and how these relate to the IN# and PR# commands.

Student References

Recommended

Books:

Apple Computer, Inc. (1983). *Applesoft BASIC programmer's reference manual.*Reading, MA: Addison-Wesley.

Apple Computer, Inc. (1983). *Apple II DOS user's manual.* Cupertino, CA: Author.

Apple Computer, Inc. (1987). *BASIC programming with ProDOS.* Reading, MA: Addison-Wesley.

Software:

Apple IIgs System Disk, ProDOS User's Disk, or *Apple II System Master (DOS 3.3).*

Student Activities:

1. Read about the Applesoft BASIC function CHR$ and the DOS and ProDOS commands IN# and PR# in one of the above references. Then, load the HELLO program you wrote to INITialize your program diskette in Activity COG-003.

   old program from Activity COG-003:

   [COG.003.1]

```
10 REM  ** HELLO PROGRAM **
20 PRINT "COMPUTER GRAPHICS"
30 PRINT "PROGRAMS"
40 PRINT: PRINT
50 PRINT "student's name"
60 PRINT "school's name"
70 PRINT "date"
80 END
```

   Add the following lines to cause the program to give the user a list of the programs on the disk each time it is booted. As you can see, line 150 uses the CHR$ function and the DOS/ProDOS command CATALOG to provide the listing.

   [COG.009.1]

```
80 REM  ***********************
90 REM  * GIVES  THE  USER    *
100 REM * A  LIST  OF  THE     *
110 REM * DISK'S  PROGRAMS *
120 REM ***********************
130 PRINT:PRINT
140 PRINT "PRESS ANY KEY TO CONTINUE THE LISTING"
150 PRINT CHR$ (4); "CATALOG"
160 END
```

If you saved this program originally with the name HELLO, use that name now to save it. Originally, you saved it as you formatted your disk with the INIT command. **Do not use the INIT command now or you will erase your disk. Use the SAVE command.**

You can use this same programming procedure and incorporate any of the DOS/ProDOS commands (RUN, LOAD, LIST, etc.) into your BASIC programs.

2. Review the INPUT command from Activity COG-005 and then key the following program into your computer. This program turns the mouse on, calls for input from the mouse, prints the values obtained from the mouse on the screen, and will stop when you press and release the mouse button.

[COG.009.2]

```
10 REM **  ACTIVITY  NINE    **
20 REM * COMPUTER GRAPHICS *
30 REM ************************
40 REM * ALLOWS FOR  VALUES *
50 REM * TO  BE  INPUT  FROM   *
60 REM * THE  APPLE  MOUSE    *
70 REM ************************
80 GOSUB 1000
90 INPUT "";X,Y,A
100 PRINT X,Y,A
110 IF A = 3 GOTO 130
120 GOTO 90
130 GOSUB 2000
140 TEXT:HOME
150 END


1000 REM ** SUBROUTINE   TO **
1010 REM ** TURN  MOUSE  ON **
1020 REM ***********************
1030 LET D$ = CHR$ (4)
1040 PRINT D$;"PR#4": PRINT CHR$ (1)
1050 PRINT D$;"PR#0"
1060 PRINT D$;"IN#4"
1070 RETURN
1080 END

2000 REM ** SUBROUTINE   TO **
2010 REM ** TURN  MOUSE  OFF *
2020 REM ***********************
2030 PRINT D$;"IN#0"
2040 PRINT D$;"PR#4":PRINT CHR$ (0)
2050 PRINT D$;"PR#0"
2060 RETURN
2070 END
```

RUN, debug, and SAVE this program with a unique name (COG.009.2).

When you run this program, it should print an endless stream of lines each with three values (see lines 90-120). The first two values will range from 0 to +1023 and the third value will be either a 1,2,3, or 4. The first of the values represents the X position of the mouse; the second, the Y position of the mouse; and the third, the status of the button on the mouse. If the third value is a 1, it means that the mouse button is down and was down during the last INPUT statement. If the third value is a 2, it means that the mouse button is down and was up during the last INPUT statement. If the third value is a 3, it means that the mouse button is up and was down during the last INPUT statement. A value of 4 means that the button is up and was up during the last INPUT statement. Try to change the values by moving the mouse around and by pressing the mouse button. If you press it and then release it in the right way you will obtain a value of 3 and the program will stop (see the IF statement at line 110).

The subroutine at lines 1000-1080 turns the mouse on. In line 1030, a CTRL D character (ASCII character #4) is assigned to the variable D$. This is to save space, since a CTRL D must precede any DOS or ProDOS command in a PRINT statement. It also makes the program readable. A CTRL D could be keyed into every print command without using the CHR$ function; however, there is nothing printed on the screen when a CRTL D is keyed. Therefore, it is invisible to someone reading your program.

At line 1040, the DOS/ProDOS command PR#4 is used to send subsequent output through slot 4, the mouse port. This is followed on the same line with a CTRL A, the command that turns the mouse on. The DOS/ProDOS command at line 1050, PR#0, directs output back to the monitor screen. The command, IN#4, at line 1060 directs future INPUT commands to take data from slot #4, the mouse port.

The subroutine that turns the mouse off (lines 2000-2070) reverses those things accomplished in the previous subroutine. The DOS/ProDOS command, IN#0, at line 1030 directs the computer to take future INPUT data from the keyboard. The two statements at line 1040 send a CRTL @ to the mouse through slot 4. This turns the mouse off. And, the line at 1050 directs output back to the monitor screen (undoing the effect of the PR#4 in the previous line).

3. Use the subroutines from the previous program, but change the main program as follows to make it a graphics program that allows the user to paint with the mouse:

[COG.009.3]

```
10 REM **   ACTIVITY  NINE   **
20 REM * COMPUTER GRAPHICS *
30 REM ************************
40 REM * ALLOWS FOR  INPUT  *
50 REM * FROM  THE  APPLE    *
60 REM * MOUSE FOR PAINTING*
70 REM ************************
75 REM * TURN  MOUSE  ON,    *
80 REM * PUT  IN  GRAPHICS   *
85 REM * MODE,  &  OBTAIN    *
90 REM * INPUT  FROM  MOUSE  *
95 REM ************************
100 GOSUB 1000
110 HGR:HCOLOR=3
120 INPUT "";X,Y,A
130 REM ************************
140 REM * CONSTRAIN  INPUT   *
```

```
150 REM * TO  FALL  WITHIN     *
160 REM * APPLE  WINDOW       *
170 REM ************************
180 IF X > 279 THEN X = 279
190 IF Y > 159 THEN Y = 159
200 REM ************************
210 REM * IF  MOUSE  BUTTON    *
220 REM * IS  DOWN  PLOT  A     *
230 REM * POINT;  DO  UNTIL     *
240 REM * MOUSE  BUTTON HAS *
250 REM * BEEN PUSHED  TWICE*
260 REM ************************
270 IF A = 1 THEN HPLOT X,Y
280 IF A = 2 AND B = 3 AND C = 2 GOTO 320
290 LET C = B:LET B = A
300 FOR W = 1 TO 100:NEXT W
310 GOTO 120
320 GOSUB 2000
330 TEXT:HOME
340 END


1000 REM ** SUBROUTINE   TO **
1010 REM ** TURN  MOUSE  ON **
1020 REM ************************
1030 LET D$ = CHR$ (4)
1040 PRINT D$;"PR#4": PRINT CHR$ (1)
1050 PRINT D$;"PR#0"
1060 PRINT D$;"IN#4"
1070 RETURN
1080 END

2000 REM ** SUBROUTINE   TO **
2010 REM ** TURN  MOUSE  OFF *
2020 REM ************************
2030 PRINT D$;"IN#0"
2040 PRINT D$;"PR#4":PRINT CHR$ (0)
2050 PRINT D$;"PR#0"
2060 RETURN
2070 END
```

Before you RUN this program, you should understand the logic of it and what is expected from the user. Lines 110-120 in the main program turn the mouse on (by calling the subroutine at 1000), put the computer in high-resolution graphics mode, set the color to white, and ask for INPUT. The subroutine at 1000 has set the computer to accept INPUT through slot #4; therefore, INPUT will come from the mouse rather than the keyboard. The quote marks in the INPUT statement are there to suppress the question mark that would appear on the screen without them.

Remember, in the previous program the values returned from the mouse ranged from 0 to +1023. If the mouse values are to be used to HPLOT on the Apple high-resolution graphics screen, they must be changed so that they do not exceed 279 for X values and 159 for Y values. This is what is being done at lines 180 and 190. This is an example of a *input constraint*. Get your instructor to explain *input constraints* in greater detail.

At line 270, the program HPLOT's a point if the mouse button is being held in the down position. Remember, if the third value (the variable A) is a 1, it means that the mouse button is down and was down during the last INPUT statement.

The function of the statements at lines 280 and 290 is to examine the status of the button during the last three INPUT's. The statements at line 290 store the values of variable A during the last two INPUT's. These values are stored in variables B and C. The IF statement at line 280 examines them along with the current value of A. When the three values indicate that the button has been pushed twice, the program will be sent to line 320 and the program will come to an END. Please note that to have the three values 2,3,and 2 returned in that order, the mouse button must be pushed (double-clicked) very quickly. If it doesn't work for you after several attempts, change the FOR..TO and NEXT loop at line 300. This is a pause loop to allow the computer to read your double click.

RUN, debug, and SAVE this program with a unique name (COG.009.3).

4. Try changing the shape of the paint brush in the previous program by changing line 270 as follows:

[COG.009.4]

from:

```
270 IF A = 1 THEN HPLOT X,Y
```

to:

```
270 IF A = 1 THEN HPLOT X,Y+4 TO X+4,Y
```

To prevent an error message because of an unacceptable X or Y value with the HPLOT command, you should also change lines 180 and 190.

from:

```
180 IF X > 279 THEN X = 279
190 IF Y > 159 THEN Y = 159
```

to:

```
180 IF X > 275 THEN X = 275
190 IF Y > 155 THEN Y = 155
```

Try other paint brushes of different sizes and shapes.

SAVE these versions of the program with unique names (COG.009.4).

5. For most graphic applications it is desirable to have a cursor appear on the screen that tracks the relative position of the mouse. Key the following main program into your computer. It contains such a routine. The routine contains a FOR..TO and NEXT loop that continually draws and then erases two lines that intersect at the current cursor position.

[COG.009.5]

```
10 REM **  ACTIVITY  NINE  **
20 REM * COMPUTER GRAPHICS *
30 REM ************************
40 REM * ALLOWS FOR INPUT  *
50 REM * FROM THE APPLE    *
60 REM * MOUSE FOR PAINTING*
70 REM ************************
80 PRINT "TO PAINT, USE MOUSE."
90 PRINT "TO QUIT, PRESS ANY"
100 PRINT "KEY ON KEYBOARD."
110 FOR W = 1 TO 9000:NEXT W
120 REM ************************
130 REM * TURN MOUSE ON,   *
140 REM * PUT IN GRAPHICS  *
150 REM * MODE, & OBTAIN   *
160 REM * INPUT FROM MOUSE *
170 REM ************************
180 GOSUB 1000
190 HGR:HCOLOR = 3
200 INPUT "";X,Y,A
210 REM ************************
220 REM * CONSTRAIN  INPUT  *
230 REM * TO  FALL  WITHIN   *
240 REM * APPLE  WINDOW      *
250 REM ************************
260 IF X < 6 THEN X = 6
270 IF X > 273 THEN X = 273
280 IF Y < 5 THEN Y = 5
290 IF Y > 154 THEN Y = 154
300 REM ************************
310 REM * DRAWS THE CURSOR *
320 REM ************************
330 FOR M = 1 TO 2
340 HPLOT X,Y-5 TO X,Y+5
350 HPLOT X-6,Y TO X+6,Y
360 HCOLOR = 0
370 NEXT M
380 HCOLOR = 3
390 REM ************************
400 REM * IF MOUSE BUTTON   *
410 REM * IS DOWN PLOT A    *
420 REM * POINT; DO UNTIL   *
430 REM * ANY KEY ON KEY-   *
440 REM * BOARD IS PRESSED  *
450 REM ************************
460 IF A = 1 THEN HPLOT X,Y
```

```
470 IF A < 0 GOTO 490
480 GOTO 120
490 GOSUB 2000
500 TEXT:HOME
510 END
```

The differences between this program and the one in Part 3 (COG.009.3) are these:

The IF..THEN statements at lines 180 and 190 were changed and two more IF ..THEN statements were added at lines 200 and 210. These changes were made to allow for the cursor dimensions (see the following change).

A routine was added (lines 300-380) to draw a cursor. The cursor is a horizontal line 12 pixels long intersecting at the midpoint with a vertical line 10 pixels long. The cursor is continually drawn (in white) and erased (redrawn in black) to coincide with each new position that is being input. If the cursor was not erased, there would be a path of cursors left across the screen similar to the paintbrush in the previous program. An HCOLOR = 3 statement was added at line 300 because when the program falls out of the loop, HCOLOR = 0 (black).

Notice that this program uses another method for the user to exit. When receiving INPUT from the mouse, a negative number in position three (our variable A) indicates that a key on the keyboard has been hit. As you can tell, this feature was used in this program to allow the user to exit by way of the keyboard (see line 470). Directions to indicate this difference were added with PRINT statements at lines 80-100. A pause loop, to allow the user time to read the directions, was added at line 110.

The statement at line 460 was changed to draw only one point when the mouse button is pushed.

The program was renumbered.

Of course, the subroutines to turn the mouse on and off are still needed. They should not be changed.

```
1000 REM ** SUBROUTINE   TO **
1010 REM ** TURN  MOUSE  ON **
1020 REM ************************
1030 LET D$ = CHR$ (4)
1040 PRINT D$;"PR#4": PRINT CHR$ (1)
1050 PRINT D$;"PR#0"
1060 PRINT D$;"IN#4"
1070 RETURN
1080 END

2000 REM ** SUBROUTINE   TO **
2010 REM ** TURN  MOUSE  OFF *
2020 REM ************************
2030 PRINT D$;"IN#0"
2040 PRINT D$;"PR#4":PRINT CHR$ (0)
2050 PRINT D$;"PR#0"
2060 RETURN
2070 END
```

RUN, debug, and SAVE this program with a unique name (COG.009.5).

Next, make the following changes in the program. These will allow the user to draw horizontal lines.

[COG.009.5.1]

change:

from:

```
80 PRINT "TO PAINT, USE MOUSE."
```

to:

```
80 PRINT "TO DRAW, USE MOUSE."
```

add:

```
185 DIM XX(2), YY(2)
195 LET N = 1
```

DELete lines 410-510 and replace with:

```
410 REM * IS  PUSHED,  RECORD *
420 REM * POINT.  AFTER  TWO  *
430 REM * PTS  ARE  INPUT,     *
440 REM * DRAW  A  LINE.  STOP*
450 REM * WHEN ANY  KEY ON  *
460 REM * KEYBOARD  IS  USED *
470 REM ************************
480 IF A = 3 THEN XX(N) = X:LET YY(N) = Y:LET N = N + 1
490 IF N = 3 THEN HPLOT XX(1),YY(1) TO XX(2),YY(1):LET N = 1
500 IF A < 0 GOTO 520
510 GOTO 120
520 GOSUB 2000
530 TEXT:HOME
540 END
```

The above program illustrates another example of an input constraint. In this program, all lines are constrained to be horizontal. Without constraints, it would be difficult to draw with a computer. You might think of constraints as the T squares and straight edges of computer graphics.

In line 480 above, several things happen if the button has been pushed on the mouse. The values of X and Y (the current cursor position) are stored in XX(1), YY(1) or XX(2), YY(2) depending on whether this is the first point input or the second. Also, the variable N is increased by increments of one. Even though there are three statements on this line and only the first is an IF..THEN, none of the statements will be executed if the IF..THEN is false. This is a unique characteristic of Applesoft BASIC. This will probably not work in the same way with other BASIC languages.

The statements at line 490 draw a line and set N back to 1, if two points have been input. As with line 480, the subsequent statement (LET N = 1) is not executed if the IF...THEN is false. Notice how YY (1), the Y value of the first point is also used as the Y value for the second endpoint. This is what keeps the line horizontal.

By making the following change the program will draw vertical lines instead of horizontal lines:

[COG.009.5.2]

from:

```
490 IF N = 3 THEN HPLOT XX(1),YY(1) TO XX(2),YY(1):LET N = 1
```

to:

```
490 IF N = 3 THEN HPLOT XX(1),YY(1) TO XX(1),YY(2):LET  N = 1
```

6. The cursor in the above program has at least two faults. It flickers too much and it acts as an eraser when you cross over an object that has been drawn. You can improve on the first fault by drawing the cursor using an Applesoft shape table and the commands, DRAW, XDRAW, and SIZE. You can do this as an optional activity. The second fault is best solved by using a machine language routine with an exclusive OR operation to draw the cursor. The latter is not a recommended activity for this course.

7. Change the program used in part 5, above so that it draws a rectangle. The program should have the user designate the diagonal corners of the box by two inputs with the mouse.

8. As an optional activity, design a program that draws circles based on points input with the mouse.

Evaluation Criteria: To evaluate the student's success in completing this activity, the teacher should give the student a two part test. The first part should be a performance test where the student is required to write a high-resolution computer graphics program incorporating DOS and ProDOS commands, especially IN# and PR#. The program should incorporate an Apple mouse or other input device. The student should be instructed to use only those commands covered in this activity or those covered in activities COG-004, COG-005, COG-006, COG-007, and COG-008. The second part should be a recall test requiring the student to define the term, input constraint.

To be considered successful, the student's program should run on the computer without error, allow input from the mouse or other device, and draw the intended image. On the second part, the student should be able to provide an acceptable definition for the term. It will be considered acceptable if it agrees with a definition found in current computer literature.

Computer Graphics Programming on an Apple II

GRAPHICS PROGRAMMING IN BASIC USING OUTPUT COMMANDS

Introduction: This activity is the seventh in a series of activities that is designed to teach the student how to draw computer images by programming an Apple computer with the BASIC language. This activity can be done on any Apple computer that uses Applesoft BASIC, the II+, IIe, IIc, or IIgs, and has a pen plotter attached (preferably a Hewlett Packard plotter or one that uses the Hewlett Packard plotting language, HP GL).

Enabling Objective: Before beginning this activity, the student should have completed Activity COG-009, Graphics Programming in BASIC Using Input Commands. He or she should have demonstrated mastery of the objectives found in that activity before beginning this one.

Performance Objective (6-12 days):

TASK/COMPETENCY--COG-010

The student should be able to write a graphics program which utilizes a graphics output device through the DOS or ProDOS commands IN# and PR# and the HP GP plotter language. The student should also be able to define the following terms relative to graphics programming:

viewport
absolute coordinates
relative coordinates
"A", "B", and "C" size paper

CONDITIONS

The above tasks should be performed under the following conditions:

a) The student should be provided an Apple computer with a disk drive, monitor, a Hewlett Packard pen plotter, an Applesoft BASIC reference manual, and the Hewlett Packard *Interfacing and Programming Manual* (for the appropriate HP plotter) when writing the graphics program.

b) The student should be expected to recall the definition of the above terms without the aid of a reference.

STANDARD OF PERFORMANCE

The above tasks should be performed to the following standards:

a) The graphics program should incorporate high-resolution graphics commands and those other Applesoft commands listed above or covered in COG-004, COG-005 COG-006, COG-007 , COG-008, or COG-009 activities. It should run without error and permit the user to input graphical data by using an Apple mouse or other input device.

b) The student should be able to supply a correct definition for all of the terms. Definitions should be considered correct if they conform to definitions found in the current computer literature.

Selected Instructional Materials: An Apple II+, IIe, IIc, or IIgs computer with printer, monitor (preferably with color), pen plotter (preferably a Hewlett Packard pen plotter), and disk drive; an Applesoft reference manual; the Hewlett Packard *Interfacing and Programming Manual* for the plotter that is to be used; and an Apple II utilities diskette or system master diskette; and a formatted diskette.

Teacher References

Recommended

Books:

Apple Computer, Inc. (1983). *Applesoft BASIC programmer's reference manual*. Reading, MA: Addison-Wesley.

Apple Computer, Inc. (1983). *Apple II DOS user's manual*. Cupertino, CA: Author.

Apple Computer, Inc. (1987). *BASIC programming with ProDOS*. Reading, MA: Addison-Wesley.

Hewlett Packard. (1983). *Interfacing and programming manual*. San Diego, CA: Hewlett Packard.

Software:

Apple IIgs System Disk, ProDOS User's Disk, or *Apple II System Master (DOS 3.3).*

Teacher Activities

1. The Applesoft BASIC library function CHR$ and the DOS and ProDOS commands IN# and PR# that were introduced in COG-009 will also be used extensively in this activity. Review these commands again, if you need to.

2. The terms viewport and window which were introduced earlier, should be reviewed again with the student.

3. Familiarize yourself with the Hewlett Packard Graphics Language (HP GL) which is explained in the *Interfacing and Programming Manual*. Read Chapters 1-5 carefully. The following HP GL mnemonic commands are important to this activity: AA, AR, CI, CP, DT, IN, IP, IW, LB, LT, PA, PD, PR, PU, SC, SI, SP, and SR. Make special note of them when reading these chapters.

4. Be certain that the plotter has been properly installed and is in working condition. The *Interfacing and Programming Manual* and the operating manual that came with your plotter should be consulted when connecting your plotter. Run a test program on the plotter before the students are ready to use it.

5. Be prepared to offer suggestions for the selection of an object to be drawn in the Student Activity Part 3. The object should require the student to plot circles and arcs.

Student References

Recommended

Books:

Apple Computer, Inc. (1983). *Applesoft BASIC programmer's reference manual*. Reading, MA: Addison-Wesley.

Apple Computer, Inc. (1983). *Apple II DOS user's manual*. Cupertino, CA: Author.

Apple Computer, Inc. (1987). *BASIC programming with ProDOS*. Reading, MA: Addison-Wesley.

Hewlett Packard. (1983). *Interfacing and programming manual*. San Diego, CA: Hewlett Packard.

Software:

Apple IIgs System Disk, ProDOS User's Disk, or Apple II System Master (DOS 3.3).

Student Activities:

1. Review again those commands introduced in COG-009--the Applesoft BASIC function CHR$ and the DOS or ProDOS commands IN# and PR#. Then, read Chapters 1-5 in the *Interfacing and Programming Manual*. Make special note of the HP GL commands IN, IP, PA, PD, PU, SC, and SP. When you feel comfortable with these commands, try them in the following program. It draws a border line on a piece of "A size" (8.5 x 11 inch) paper. Therefore, before you RUN this program, be sure you are working at a computer that has a Hewlett-Packard plotter properly attached. You will also need to turn the plotter on, place pens in the pen holders, and have a sheet of 8.5" x 11" ("A size") paper properly loaded.

COG.010.1

```
10 REM **    ACTIVITY TEN    **
20 REM * COMPUTER GRAPHICS *
30 REM ************************
40 REM * PLOTS A BORDER      *
50 REM * USING THE HEWLETT *
60 REM * PACKARD PLOTTER    *
70 REM ************************
80 REM * SELECT SLOT FOR     *
90 REM * PLOTTER OUTPUT      *
95 REM ************************
100 LET D$ = CHR$ (4)
110 PRINT D$; "PR#2"
120 REM ************************
130 REM * SEND PLOTTER CMDS*
140 REM * TO INITIATE, AND    *
150 REM * TO SET VIEWPORT     *
160 REM * AND WINDOW          *
170 REM ************************
180 PRINT "IN;"
190 PRINT "IP 0,0,10100,7140;"
200 PRINT "SC 0,10,0,7.5;"
210 REM ************************
220 REM * SEND PLOTTER CMDS*
230 REM * TO SELECT PEN,      *
240 REM * PLOT ABSOLUTE, &   *
250 REM * PLOT BORDER W/PEN*
260 REM * UP AND PEN DOWN    *
270 REM ************************
280 PRINT "SP 1;"
290 PRINT "PA,PU 0,0 PD 9.8,0 9.8,7 0,7 0,0;"
600 REM ************************
610 REM * RETURN PEN TO       *
620 REM * HOLDER & CEASE     *
630 REM * OUTPUT TO SLOT # 2 *
640 REM ************************
650 PRINT "SP 0;"
660 PRINT D$;"PR#0"
670 END
```

RUN, debug, and SAVE this program with a unique name (COG.010.1).

In the above program, output to the plotter is directed through slot #2. If your plotter is connected to a card in a different slot, you will need to change the PR# command in line 110 accordingly. Plotter commands begin at line 180 with the command to initiate the plotter and cease at line 650 with the command to select pen 0 (return the pen to its holder). Notice that these commands and all intermediate plotter commands end with a semi-colon. If they did not, a carriage return would be output and the plotter commands would be terminated. The IP command at line 190 is used to select the viewport, and the SC command at line 200 is used to select the window. The same considerations that have been used in previous activities when selecting a viewport and a window need to be used here, that is, the aspect ratios of the viewport and the window need to be the same to avoid distortion of the image being drawn.

2. Read about the HP GL commands DT, LB, LT, PR, and SR. Using the above program, COG.010.1, as a skeleton program, add the necessary commands using lines beginning with 300 but smaller than 600 to draw the image shown below.

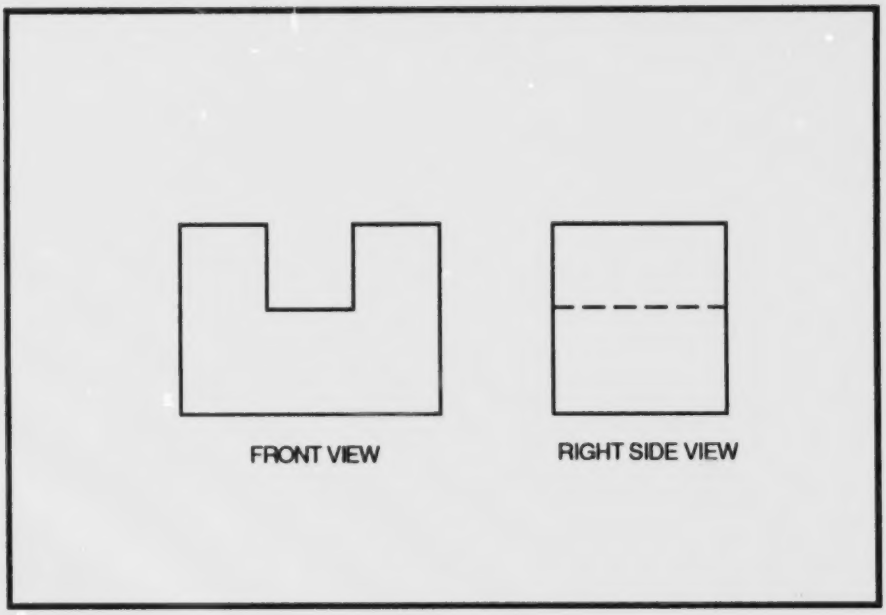


Figure 11.--Two View Drawing

[COG.010.2]

```
10 REM **   ACTIVITY TEN    **
20 REM * COMPUTER GRAPHICS *
30 REM ************************
40 REM * PLOTS TWO VIEWS    *
50 REM * USING THE HEWLETT *
60 REM * PACKARD PLOTTER    *
70 REM ************************
80 REM * SELECT SLOT FOR    *
90 REM * PLOTTER OUTPUT     *
95 REM ************************
100 LET D$ = CHR$ (4)
110 PRINT D$; "PR#2"
120 REM ************************
130 REM * SEND PLOTTER CMDS*
140 REM * TO INITIATE, AND   *
150 REM * TO SET VIEWPORT    *
160 REM * AND WINDOW         *
170 REM ************************
180 PRINT "IN;"
190 PRINT "IP 0,0,10100,7140;"
200 PRINT "SC 0,10,0,7.5;"
210 REM ************************
```

```
220 REM * SEND PLOTTER  CMDS*
230 REM * TO  SELECT  PEN,       *
240 REM * PLOT  ABSOLUTE, &    *
250 REM * PLOT  BORDER   W/PEN*
260 REM * UP  AND  PEN  DOWN   *
270 REM ************************
280 PRINT "SP 1;"
290 PRINT "PA,PU 0,0 PD 9.8,0 9.8,7 0,7 0,0;"
300
310
  .
  .
  .
  .
  .

600 REM ************************
610 REM * RETURN  PEN  TO       *
620 REM * HOLDER  &  CEASE     *
630 REM * OUTPUT  TO  SLOT # 2 *
640 REM ************************
650 PRINT "SP 0;"
660 PRINT D$;"PR#0"
670 END
```

RUN, debug, and SAVE this program with a unique name (COG.010.2).

3. Read about the HP GL commands AA, AR, and CI. Modify the previous programs, COG. 010.1 or COG.010.2, to plot views of an object that contains circles and arcs. With your instructor's help, select such an object from a drafting text or workbook.

[COG.010.3]

```
10 REM **   ACTIVITY  TEN    **
20 REM * COMPUTER GRAPHICS *
30 REM ************************
40 REM * PLOTS  ARCS/CIRCLES *
50 REM * USING THE  HEWLETT *
60 REM * PACKARD  PLOTTER    *
70 REM ************************
80 REM * SELECT  SLOT  FOR     *
90 REM * PLOTTER  OUTPUT      *
95 REM ************************
100 LET D$ = CHR$ (4)
110 PRINT D$; "PR#2"
120 REM ************************
130 REM * SEND PLOTTER  CMDS*
140 REM * TO  INITIATE,  AND    *
150 REM * TO  SET  VIEWPORT    *
160 REM * AND  WINDOW          *
170 REM ************************
180 PRINT "IN;"
190 PRINT "IP 0,0,10100,7140;"
```

```
200 PRINT "SC 0,10,0,7.5;"
210 REM ************************
220 REM * SEND PLOTTER CMDS*
230 REM * TO SELECT PEN,       *
240 REM * PLOT ABSOLUTE, &    *
250 REM * PLOT BORDER W/PEN *
260 REM * UP AND PEN DOWN   *
270 REM ************************
280 PRINT "SP 1;"
290 PRINT "PA,PU 0,0 PD 9.8,0 9.8,7 0,7 0,0;"
300
310
 .
 .
 .
 .

600 REM ************************
610 REM * RETURN PEN TO       *
620 REM * HOLDER & CEASE     *
630 REM * OUTPUT TO SLOT # 2 *
640 REM ************************
650 PRINT "SP 0;"
660 PRINT D$;"PR#0"
670 END
```

RUN, debug, and SAVE this program with a unique name (COG.010.3).

4. For an optional activity, develop an interactive program that utilizes the plotter and the mouse. The program could allow the user to draw an object on the screen with the mouse and then with a suitable command, output the drawing to the plotter. Before you begin this project, you may want to review the use of variables with HP GL commands, and the use of arrays to store drawing data for later use.

Evaluation Criteria: To evaluate the student's success in completing this activity, the teacher should give the student a two part test. The first part should be a performance test where the student is required to write a high-resolution computer graphics program incorporating DOS and ProDOS commands, especially IN# and PR# and the HP GL mnemonic commands. The program should incorporate a Hewlett Packard pen plotter or other suitable output device. The student should be instructed to use only those commands covered in this activity or those covered in activities COG-004, COG-005, COG-006, COG-007, COG-008, and COG-009. The second part should be a recall test requiring the student to define the terms, viewport, window, absolute coordinates, relative coordinates, and "A", "B", and "C" size paper.

To be considered successful, the student's programs should run on the computer without error, allow output to the plotter, and plot the intended image. On the second part, the student should be able to provide acceptable definitions for the terms. They will be considered acceptable if they agree with definitions found in current computer literature.

Microcomputer Graphics Applications
COMPUTER GRAPHICS PAINTING

Introduction: Computer paint systems are interactive computer graphics systems used to create 2D computer images that simulate the characteristics of an image painted on canvas or paper. These systems provide the user with the computer equivalent of a canvas, paint brush, air brush (or spray can), pencil, straightedge, eraser, type, and much more. While these systems have limitations when compared to the conventional method of painting, they are less messy and in many ways, easier to experiment with.

Enabling Objective: Before beginning this activity, the student should have completed Activity COG-003, How to Operate the Apple Computer. He or she should be able to turn the computer off and on, be familiar with the keyboard, be able to load and save programs to the disk drive, and be able to use the printer. If students are to work through both this activity and COG-013, Computer Graphics Animation, or COG-014, Desktop Publishing, this activity should be done first.

Performance Objective (15-20 days):

TASK/COMPETENCY--COG-011

Upon completion of this activity, the student will be able to describe a computer paint system application and use a system to paint computer generated images. The student will also be able to recall definitions for the following:

palette	shade
drag	shadow
cut and paste	pixel
fill pattern	fat bits
fill pattern leak	basic colors of light

CONDITIONS

The above tasks will be performed under the following conditions:

a) The terms will be recalled from memory.

b) The student will be provided resource materials when she or he is asked to write a description of or describe orally an application.

c) To generate the painted image, the student should be provided a outline (drawn on paper) of the 2D image to be painted; an Apple IIe (with 128K) or the IIc computer system with at least one disk drive (2 are suggested); a mouse, Apple tablet, KoalaPad, or joystick; a color monitor; and the software package *Dazzle Draw*; or an Apple IIgs and either the software package *Paintworks* or *Dazzle Draw*.

STANDARDS OF PERFORMANCE

The above tasks should be performed to the following standards:

a) The student will be expected to provide a definition (similar to that found in the current computer graphics literature) for 3 out of 4 terms selected randomly.

b) The description of an application should include the advantages of using computer paint systems over conventional techniques, the types of computers used, careers associated with the application, and some specific examples of work accomplished with this application.

c) The 2D painted image will require the use of at least five paint system functions. The painted image will compare favorably with a painted image prepared by the instructor.

Selected Instructional Materials: An Apple IIe, IIc, or IIgs computer with printer; color monitor; disk drive (preferably two); and a mouse, Apple tablet, KoalaPad, or joystick; and the application program *Blazing Paddles, Dazzle Draw, DeluxePaint II,* or *Paintworks Plus*. Note: *DeluxePaint II* and *Paintworks Plus* will only run on an Apple IIgs. Also needed are the manuals or user's guides which are provided with the software packages.

Teacher References

Recommended

Books: The user's guide or user's manual that is supplied with one of the software packages listed below.

Videotape: Frost and Sullivan, Inc. (1987). *Graphic arts systems/video text report (VTR) 002*. A Frost and Sullivan video text report. Available from Frost and Sullivan, Department R-1, 106 Fulton Street, New York, NY 10038.

Software: One of the following:

Darooge, M., McCullough, E. & Walker, D. (198_). *Blazing Paddles*. Available from Baudville, 1001 Medical Park Drive S.E., Grand Rapids, MI 49506 or a local computer software dealer.

Silva, D. (198_). *DeluxePaint II*. Available from Electronic Arts, 1820 Gateway Drive, San Mateo, CA 94404 or a local computer software dealer.

Snider, D. (1984,1987). *Dazzle Draw*. Available from Broderbund Software, Inc., P.O. Box 12947, San Rafael, CA 94913-2947 or a local computer software dealer.

Version Soft, Inc. (1986). *Paintworks Plus*. Available from Activision, *Inc.*, *P.O. Box* 7287, Mountain View, CA 94039 or a local Apple computer dealer.

Suggested

Books:

Kemnitzer, R.B. (1983). *Rendering with markers*. New York: Watson-Guptill Publications.

Martin, C.L. (1968). *Design graphics*. New York, NY: The Macmillan Company.

Ungar, J. (1985). *Rendering in mixed media*. New York, NY: Watson-Guptill Publications.

Teacher Activities

1. Familiarize yourself with computer paint systems by reading the guide that accompanies the paint program you have chosen for your students to use and by using the paint program.

2. Using your favorite drafting textbook, review the following and prepare demonstrations for your students. Although you can use other media (such as the blackboard) to demonstrate, use the computer when possible:

   Multiview drawings

   Shape description.
   Reproducing irregular shapes with grids

   Pictorial drawing

   Isometric drawings
   Perspective drawings

   Using your favorite rendering textbook, review the following and prepare demonstrations for your students. You should emphasize how the basic geometric shapes are rendered (cubes, cylinders, cones, spheres, and pyramids):

   Rendering techniques

   Shades in black and white
   Shadows in black and white
   Shades and Shadows in color

3. If time permits, work through the student activities.

4. Prepare for and demonstrate the software package you plan for your students to use.

5. Be prepared to offer suggestions for Part 8 of the student activities. For some students who have difficulty drawing and painting objects or determining what they should draw and paint, you may need to give them objects to paint (give them drawings similar to what you would find in a coloring book and let them paint them). However, this should be done only as a last resort.

6. When showing the videotape, *Graphic Arts Systems/Video Text Report (VTR) 002*, you should skip (fast forward over) those parts that are highly technical and those parts that deal with solid modeling. The parts of the video that are important to this activity are those parts that show 2D paint systems.

Student References

Recommended

Books: The user's guide or user's manual that is supplied with one of the following software packages.

Videotape: Frost and Sullivan, Inc. (1987). *Graphic arts systems/video text report (VTR) 002*. A Frost and Sullivan video text report. Available from Frost and Sullivan, Department R-1, 106 Fulton Street, New York, NY 10038.

Software: One of the following:

Darooge, M., McCullough, E. & Walker, D. (198_). *Blazing Paddles*.

Silva, D. (198_). *DeluxePaint II*.

Snider, D. (1984,1987). *Dazzle Draw*.

Version Soft, Inc. (1986). *Paintworks Plus*.

Suggested

Books:

Kemnitzer, R.B. (1983). *Rendering with markers*. New York: Watson-Guptill Publications.

Martin, C.L. (1968). *Design graphics*. New York, NY: The Macmillan Company.

Ungar, J. (1985). *Rendering in mixed media*. New York, NY: Watson-Guptill Publications.

Student Activities

1. Observe your instructor as he or she demonstrates the techniques for representing objects with multiview, isometric, and perspective sketches and drawings.

2. Try using these techniques to sketch simple objects with paper and pencil.

3. As the instructor makes future demonstrations and when you view the videotape in Part 4 below, observe how the following terms are used:

palette	shade
drag	shadow
cut and paste	pixel
fill pattern	fat bits
fill pattern leak	basic colors of light

   If you do not understand any of these terms, be sure and ask your instructor for a definition.

4. View those parts of the videotape, *Graphic Arts Systems/Video Text Report (VTR) 002*, that deal with 2D paint systems.

5. Observe your instructor as he or she demonstrates how objects are made to look three dimensional by using shades and shadows.

6. Try these techniques by sketching spheres, cubes, pyramids, cylinders, and cones with surfaces shaded and shadows plotted. Sketch the objects with paper and pencil. Next, try to sketch the same objects with shades and shadows drawn in color with magic marker or colored pencil.

7. Observe your instructor as he or she demonstrates the use of a computer paint system.

8. Practice using the paint program:

   a) Read the user's guide.

   b) Practice using the different functions.

   c) When you think you are ready, use the paint program to draw and paint a simple flat object or design. The following are offered as suggestions:

      (1) a vanity license plate for an automobile or a bicycle.

      (2) a personal "T" shirt design.

      (3) a school sports emblem for a football helmet, baseball cap, or "T" shirt.

(4) a design for the cover of a wall poster, notebook, school play program, sports event program, or school newspaper masthead.

(5) a logo for your personal calling card or one for a parent.

(6) a logo for a fictitious company.

d) Next, use the paint program to draw and paint a single multiview of an object, such as:

(1) a front view of a small house or a business entrance.

(2) the side view of car.

(3) a front view of your computer.

(4) a top view of the family house and lot with future landscaping designs.

(5) a front or a side view of a piece of furniture.

With the above multiviews, you can begin to incorporate shadows and shading to show some depth, for example, show shadows around doors and windows in the front view of the house.

e) Use the paint program to draw and paint the pictorial drawings of the geometric shapes you drew in Part 5 above.

f) Lastly, use the paint program to draw and paint objects and scenes using isometrics and perspectives. Use shades and shadows to help show depth and project realism. Following are some ideas for this exercise:

(1) an isometric or perspective of a model car design.

(2) a perspective of a small house designed by you.

(3) a new design for the interior of your bedroom.

(4) an isometric of a machine part, tool, or other object found in your technology education laboratory.

9. As an optional activity, create a painting with your paint system and enter it in one of the many computer art contests for high school students. Magazines such as *A+* and *Classroom Computer Learning* announce contests such as this from time to time.

Evaluation Criteria: The student should be given a short quiz where she or he is to define four randomly selected terms from this unit. Her or his definitions should conform to definitions found in the recommended or suggested texts or videotapes. She or he should be able to define three of four randomly selected terms successfully.

The student should also be evaluated on how well she or he can describe a computer paint application. The student's description of an application should include the advantages of using computer painting over conventional techniques, the types of computers used, careers associated with the application, and some specific examples of work accomplished with this application. Students should be given extra credit if their examples are not like those described in the videotape or the text (required additional research), and if they demonstrate an exceptional amount of enthusiasm in their presentation.

The third and last part of the evaluation should be based on the student's creativity, technique, and enthusiasm in painting images with the paint program. Special attention should be given to Student Activity, Part 8. High marks should be given for creativity when the student is able to develop a number of different ideas for paintings, when the paintings are unlike those of other students in the class, and when the paintings contain elaborate detail. High marks for technique should be given when the student has correctly plotted shadows, surfaces are correctly shaded, and highlighting enhances the 3D effect. Extra credit should also be given for scenes involving human figures or parts of the human anatomy.

Microcomputer Graphics Applications
COMPUTER AIDED DESIGN

Introduction: Computer aided design (CAD) systems are interactive computer graphics systems used to create 2D or 3D drawings of topographical maps, mechanical or electrical devices, architectural structures, and other entities designed or altered by engineers, architects, and designers. This computer graphics application has drastically changed the way that we design items and the way they are graphically represented.

Enabling Objective: Before beginning this activity, the student should have completed Activity COG-003, How to Operate the Apple Computer. He or she should be able to turn the computer off and on, be familiar with the keyboard, be able to load and save programs to the disk drive, and be able to use the printer.

Performance Objective (15-20 days):

TASK/COMPETENCY--COG-012

Upon completion of this activity, the student should be able to describe a 2-D computer aided drafting system application and use a system to produce technical drawings. He or she should also be able to recall definitions for the following:

CAD	wireframe
digitizing	window
grid	viewport
resolution	Cartesian coordinates
origin	isometric drawings
orthographic drawings	relative coordinate
layers	perspective drawings
polyline	zoom
mirror	cursor
CAD function	jaggies
clipping	input
output	interactive
AEC	CIM
IGES	absolute coordinate

CONDITIONS

The above tasks will be performed under the following conditions:

a) The terms will be recalled from memory.

b) The student will be provided resource materials when she/he is asked to write a description of or orally describe an application.

c) The student will be provided the object they are to make a drawing of or a complete graphical description. An acceptable practice is to provide them a drawing in a different form than that which is expected (e.g., an isometric when a multiview is expected). The student will also be provided an Apple IIe, IIc, or IIgs computer system with at least one disk drive (2 are suggested); a mouse, Apple tablet, KoalaPad, or joystick; a monitor; and a CAD software package.

STANDARDS OF PERFORMANCE

The above tasks should be performed to the following standards:

a) The student will be expected to provide a definition (similar to that found in the current computer graphics literature) for 7 out of 10 randomly selected terms.

b) The description of an application should include the advantages of using computer aided drafting (CAD) systems over conventional techniques, the types of computers used, careers associated with the application, and some specific examples of work accomplished with this application.

c) The drawing will require the use of at least five CAD functions. The drawing will be correct and conform to standard drafting conventions as described in the ANSI drafting standards.

Selected Instructional Materials: An Apple IIe, IIc or IIgs computer with printer; monitor; disk drive (preferably two); and a mouse, Apple tablet, KoalaPad, or joystick; a CAD application program; and the documentation provided with the application program.

Teacher References

Recommended

Books:

The user's guide or user's manual that is supplied with the software package that has been selected for use in this activity, and one of the following:

Bertoline, G.R. (1985). *Fundamentals of CAD*. Albany, NY: Delmar Publishing, Co.

Goetsch, D.L. (1985). *Computer-aided drafting*. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Voisinet, D.D. (1986). *Introduction to computer-aided drafting (2nd edition)*. New York, NY: Gregg Division of McGraw-Hill.

Videotapes:

CIM Solutions. (1986). *Computer aided design: a video evaluative report I*. Available from CIM Solutions, University Station, P.O. Box 7041, Provo Utah 84602.

CIM Solutions. (1986). *Computer aided design: a video evaluative report II*. Available from CIM Solutions, University Station, P.O. Box 7041, Provo Utah 84602.

Frost and Sullivan, Inc. (1987). *Workstations/video text report (VTR) 001*. A Frost and Sullivan video text report. Available from Frost and Sullivan, Department R-1, 106 Fulton Street, New York, NY 10038.

Silicon Graphics Computer Systems. (1987). *Superworkstation demonstration*. Silicon Graphics, Corporate Headquarters, 2011 Suerlin Road, Mountain View, CA 94043.

Software:

One of the following:

Langer, V. (1985). *MATC-CAD* Available from the Milwaukee Area Technical College, 1015 North Sixth Street, Milwaukee, Wisconsin 53203.

Versacad Corporation. (1987). *Entry Level CADApple*. Available from Versacad Corporation, 7372 Prince Drive, Huntington Beach, CA 92647 or a local computer software dealer.

Suggested

Books:

Lamit, L. G., & Paige, V. (1987). *Computer-aided design and drafting*. Columbus, OH: Merrill.

McKissick, M. L. (1987). *Computer-aided drafting and design*. Englewood Cliffs, NJ: Prentice-Hall.

Voisinet, D.D. (1987). *Computer-aided drafting and design, concepts and applications*. New York, NY: Gregg Division of McGraw-Hill.

Teacher Activities

1. Familiarize yourself with computer aided design systems and terminology by reading the text you selected from the recommended list above or one of the suggested texts.

2. Preview the videotapes carefully. You may find that they are not suitable for your students. This is especially true of the Frost and Sullivan tape, *Workstations/Video Text Report (VTR) 001*. It is a very technical tape on state-of-the-art workstations. If you do show it to your students, be sure to discuss terminology with them beforehand. The two CIM Solutions tapes should not be shown in their entirety. You should select the demonstration sequence for only one or two of the systems. The Silicon Graphics is a short tape and is probably the best for showing future directions in CAD.

3. Familiarize yourself with the CAD software you have selected for this activity.

4. If time permits, work through the student activities.

5. Prepare to demonstrate the CAD package you plan for your students to use.

6. Select drafting problems for the students to do in Part 4 of the Student Activities. The following will help guide you in making those selections:

For:	Type of problem suggested:
4,b	Figures 6-42 through 6-47 in *Fundamentals of CAD* by Bertoline. Select a problem like one of these and have them draw a multiview without dimensions.
4,c),(1)	Figures 6-38 or 6-39 in *Fundamentals of CAD* by Bertoline. Select a problem like one of these and have the students draw an isometric without dimensions.
4,c),(2)	Figures 6-34 or 8-36 in *Fundamentals of CAD* by Bertoline. Select a problem like one of these and have the students draw a fully dimensioned multiview.
4,c),(3)	Figures 9-38, 9-40, and 9-42 in *Fundamentals of CAD* by Bertoline. Select a problem like these. Have the students draw a set of electrical symbols similar to those in Figure 9-38 and then use them to draw a schematic similar to Figure 9-40 or 9-42.
4,c),(4)	Figure 9-39. Have students draw architectural symbols similar to those in this figure and then use them to draw a simple architectural structure of their own design.

7. Plan a field trip to a local engineer's, or architect's office or to a local business or industry that uses a CAD system. Refer to Acitivity COG-002.

Student References

Recommended

Books:

The user's guide or user's manual that is supplied with the software package that has been selected for use in this activity, and one of the following:

Bertoline, G.R. (1985). *Fundamentals of CAD*. Albany, NY: Delmar Publishing, Co.

Goetsch, D.L. (1985). *Computer-aided drafting*. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Voisinet, D.D. (1986). *Introduction to computer-aided drafting (2nd edition)*. New York, NY: Gregg Division of McGraw-Hill.

Videotapes:

CIM Solutions. (1986). *Computer aided design: a video evaluative report I*.

CIM Solutions. (1986). *Computer aided design: a video evaluative report II*.

Frost and Sullivan, Inc. (1987). *Workstations/video text report (VTR) 001*.

Software:

One of the following:

Langer, V. (1985). *MATC-CAD*.

Versacad Corporation. (1987). *Entry Level CADApple*.

Suggested

Books:

Lamit, L. G., & Paige, V. (1987). *Computer-aided design and drafting*. Columbus, OH: Merrill.

McKissick, M. L. (1987). *Computer-aided drafting and design*. Englewood Cliffs, NJ: Prentice-Hall.

Voisinet, D.D. (1987). *Computer-aided drafting and design, concepts and applications*. New York, NY: Gregg Division of McGraw-Hill.

Student Activities

1. Read the introductory chapters in the textbook your instructor has selected from the list of recommended books shown above. When you have completed your reading, you should be able to recall the definitions for each of the following terms:

CAD	wireframe
digitizing	window
grid	viewport
resolution	Cartesian coordinates
origin	isometric drawings
menu	orthographic drawings
layers	perspective drawings
polyline	zoom
mirror	cursor
CAD function	jaggies
clipping	input
output	interactive
AEC	CIM
IGES	absolute coordinates
relative coordinates	

2. If your instructor recommends it, view those portions of the videotapes she or he selects.

3. Observe your instructor as she or he demonstrates the computer aided design (CAD) application program that has been selected for your use in this activity.

4. Practice using the CAD application program:

   a) Read the user's guide, reference manual, or student workbook that accompanies the software package.

   b) Use the CAD software package to produce a multiview (2 or 3 view) drawing of a simple object without dimensions. Your instructor will supply you with a problem.

   c) Continue to learn more about the CAD software by producing the following drawings:

      (1) an isometric drawing of a simple object.

      (2) a multiview drawing of a simple object fully dimensioned.

      (3) a schematic drawing of a simple electrical circuit.

      (4) a floor plan a of simple architectural structure (something similar to a child's playhouse).

7. Prepare a list of questions to ask those CAD professionals you will visit on the field trip that your instructor has planned for you and your class. You may want to learn more about their professional training, the types of drawings they use their CAD system for, the type of system they use, etc.

8. After returning from your field trip, write a short paper explaining how CAD systems are used in industry. How did the professionals you visited use CAD systems to increase productivity? What kinds of output do they generate? How are CAD professionals trained? What are some of the problems encountered when implementing CAD into an operation?

Evaluation Criteria: The student should be given a short quiz where she/he is to define ten randomly selected terms from this unit. Her/his definitions should conform to definitions found in the recommended or suggested texts or videotapes. She/he should be able to define seven of the ten terms successfully.

The student should also be evaluated on how well she/he can describe a computer aided design application (see Student Activities, Part 8). The student's description of an application should include the advantages of using computer aided design over conventional techniques, the types of computers used, careers associated with the application, and some specific examples of work accomplished with this application. Students should be given extra credit if their examples are not like those described in the videotape, the text, or on the field trip (it required additional research), and if they demonstrate an exceptional amount of enthusiasm in their presentation.

The third and last part of the evaluation should be based on the student's skill and enthusiasm in generating drawings with the CAD application. Special attention should be given to Student Activity, Part 4, c). The drawings generated should be correctly drawn and dimensioned (when dimensioning is called for) according to ANSI drafting standards. The student should also demonstrate a certain amount of speed and the ability to work without help from the teacher or other students.

Microcomputer Graphics Applications
COMPUTER AIDED ANIMATION

Introduction: Animation, the art of making drawings that appear to move, used to require expensive movie making equipment and loads of artistic talent. While artistic talent is still required, computers have simplified the task tremendously. Now, most anyone with a personal computer, can do simple animation. Although the animation created on a personal computer is not as eloquent as that which is produced in the Walt Disney Studio or with large super computers, it does approach that which is used in arcade game animation.

Computer generated animation is finding applications in many new fields of endeavor. The computer is being used to generate animation sequences in the traditional areas of movie making, cartoons for television, arcade games, and television commercials. New applications are also being found in flight and space simulation, education, medical research, engineering, sports, chemistry, and biology. In most of these applications, computer animation is used to simulate the physical laws of nature. It is now much more practical for these new applications, because the computer has taken over much of the laborious task of drawing hundreds of animation cells (24-30 per second of animation) needed to create a few minutes of film or videotape.

Enabling Objective: Before beginning this activity, the student should have completed Activity COG-003, How to Operate the Apple Computer. He or she should be able to turn the computer off and on, be familiar with the keyboard, be able to load and save programs to the disk drive, and be able to use the printer. If students are to work through both this activity and Activity COG-011, Computer Graphics Painting, Activity COG-011 should be done first.

Performance Objective (15-20 days):

TASK/COMPETENCY--COG-013

Upon completion of this activity, the student will be able to describe a computer generated animation application, develop a storyboard, and use the storyboard and a computer system to produce a 2D animation sequence. The student should also be able to:

flicker
pixel
object blend sequence
flip book
storyboard
persistence of vision
frame buffer
real time animation
thaumatrope

CONDITIONS

The above tasks will be performed under the following conditions:

a) The animation terms will be recalled from memory.

b) The student will be provided resource materials when she or he is asked to write a description of or describe orally an application.

c) To generate the animation sequence, the student should be provided a series of animation cells drawn on paper and an Apple IIe (with 128K) or the IIc computer system with at least one disk drive (2 are suggested); a mouse, Apple tablet, KoalaPad, or joystick; and the software package *Animate*; or an Apple IIgs and the software package *Paintworks Plus* or *Panitworks Gold.*

STANDARDS OF PERFORMANCE

The above tasks should be performed to the following standards:

a) The student will be expected to provide a definition (similar to that found in the current computer graphics literature) for 3 out of 4 terms selected randomly.

b) The description of an application should include the advantages of using computer animation over conventional techniques, the types of computers used, careers associated with the application, and some specific examples of work accomplished with this application.

c) The animated sequence will contain at least six cells and represent a smooth fluid movement of the image. The storyboard and individual cells will be skillfully drawn (will compare favorably to an example prepared by the instructor).

Selected Instructional Materials: An Apple IIe or IIc computer with printer; monitor (preferably with color); disk drive (preferably two); and a mouse, Apple tablet, KoalaPad, or joystick; and the application program *Animate*; or the Apple IIgs and *Paintworks Plus or Paintworks Gold.* Also needed are the references *Animation, The Animation Book,* and *Computer Animation Primer.*

Teacher References

Recommended

Books:

Fox, D., & Waite, M. (1984). *Computer animation primer.* New York: McGraw-Hill.

Blair, P. (1986). *Animation.* Tustin, CA: Foster Art Service Inc.

Laybourne, K. (1979). *The animation book.* New York: Crown.

Paintworks plus user's guide or *Animate user's manual.* These are supplied with the software.

Videotapes:

Dupont, C. & Cinemagic Productions. (1987). *Computer magic.*

Electronic Pictures Corporation. (1987). *Computer graphic special on TV commercials and animation.*

National Computer Graphics Association. (1988). *1988 International computer animation competition winners*.

Software: *Paintworks plus* by Activision (Apple IIgs only) or *Animate* by Broderbund Software.

Teacher Activities

1. Familiarize yourself with computer animation by reading the following:

   a) Part One of *Computer Animation Primer* by Fox and Waite.

   b) *Animation* by Preston Blair.

   c) *The Animation Book* by Kit Laybourne.

2. Preview the videotapes *Computer Magic, Computer Graphic Special on TV Commercials and Animation,* and *1988 International Computer Animation Competition Winners.*

3. Familiarize yourself with the software package *Animate* or *Paintworks Plus.* If you have a IIgs, *Paintworks Plus* is suggested.

4. If time permits, work through the student activities.

5. Prepare a demonstration to demonstrate the software package you plan for your students to use.

6. Be prepared to offer suggestions for Part 10 of the student activities. For those students who have difficulty with this part, be prepared to provide a lot of encouragement and as a last resort, a sequence that they can copy.

Student References

Recommended

Books: Fox, D., & Waite, M. (1984). *Computer animation primer.* New York: McGraw-Hill.

Blair, P. (1986). *Animation.* Tustin, CA: Foster Art Service Inc.

Paintworks plus user's guide or *Animate user's manual.* These are supplied with the software.

Videotapes: Dupont, C. & Cinemagic Productions. (1987). *Computer magic.*

Electronic Pictures Corporation. (1987). *Computer graphic special on TV commercials and animation.*

National Computer Graphics Association. (1988). *1988 International computer animation competition winners.*

Software: *Paintworks plus* by Activision (Apple IIgs only) or *Animate* by Broderbund Software.

Suggested

Books: Laybourne, K. (1979). *The animation book.* New York: Crown.

Student Activities

1. View the videotape, *Computer Magic.*

2. Read Chapter One in *Computer Animation Primer.* As you read, make a list of new terms that are associated with computer animation. When you are through, you should be able to define the terms:

flicker	persistence of vision
pixel	frame buffer
object blend sequence	real time animation
flip book	thaumatrope
storyboard	

3. Scan through the book *Animation.* Try sketching some of the figures found in this book. If it is available, scan through *The Animation Book.*

4. View the videotape, *Computer Graphic Special on TV Commercials and Animation.*

5. Observe your instructor as she or he demonstrates the *Animate* or *Paintworks Plus* application program.

6. Practice using either the *Animate* or *Paintworks Plus* application program:

   a) Read the user's guide that accompanies the software package.

   b) Use the software package to design a simple animation sequence. A suggested sequence is that of a bouncing ball.

      (1) Sketch a five cell storyboard. Draw a ball in four suspended positions, one above the other and equally spaced, and for the fifth cell, draw the ball colliding with the floor.

      (2) Now draw the five cells on the computer. Hint: use the zoom and copy features of the software. Make the ball have a three dimensional effect by adding highlights and shading. Realism can also be enhanced by adding a shadow on the floor. You may even want to draw background scenery--a basketball or volleyball court.

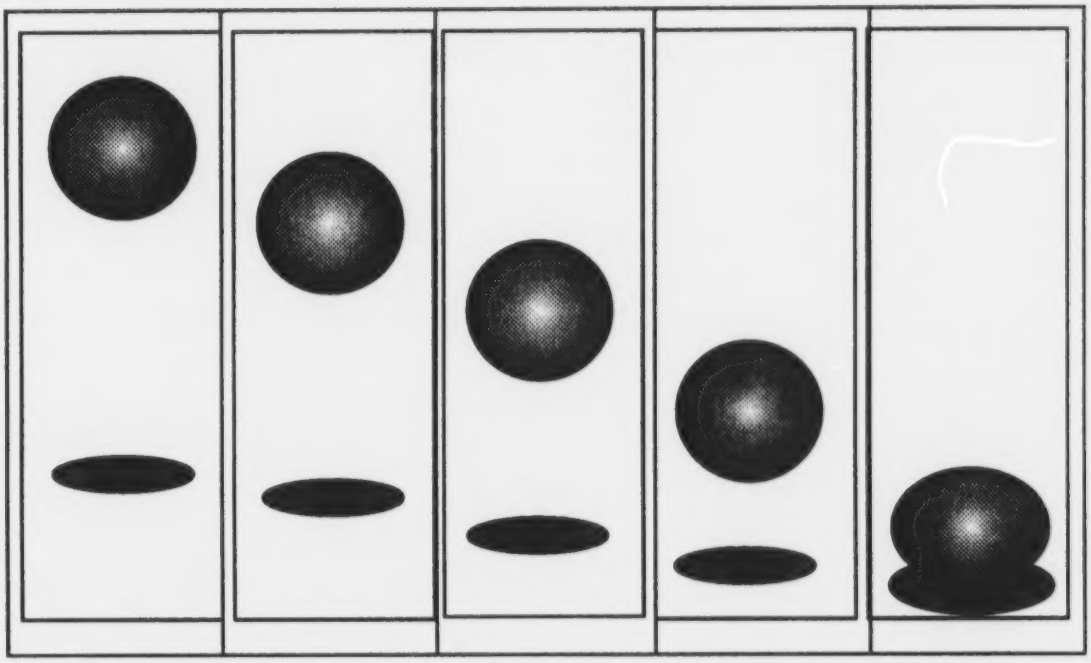


Figure 12.--Five Animation Cells of a Bouncing Ball

(3) When you have finished drawing all of the cells, animate the sequence by stepping through the cells from the ball's highest position to the collision cell and then back through the cells in reverse order to the highest position. When finished, you should have a nine cell animation sequence.

(4) Be sure to save your animation sequence on a diskette. If the proper type of video recording equipment is available, record your sequence on videotape.

7. View the videotape, *Computer Graphic Special on TV Commercials and Animation*. When you finish, write a short report on "Computer Generated Animation." In the report, give a brief description of how it is used (give specific examples), the types of computers used, the types of careers associated with this field, and tell how computers have benefitted the art of animation. When doing the research for this report be sure to consult with your librarian for additional books on the subject. You may also want to write your report using a computer and a word processing program.

8. Animating a ball and animating a human or animal figure require different techniques. Using the software you used to animate the ball, draw the eight cells shown below of the human stick figure. After you have the cells drawn, animate the figure. Next, add your own background. If your software permits, add sound. Be sure to save your sequence on disk.

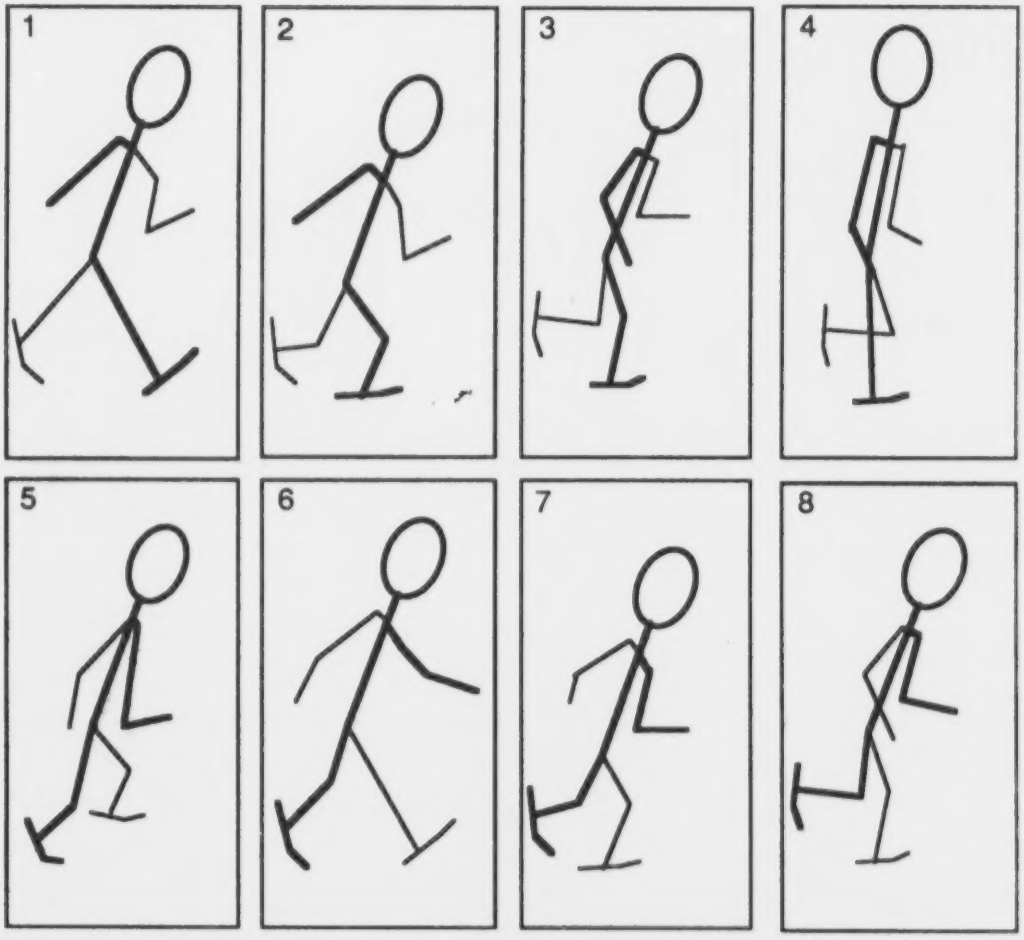


Figure 13--Eight Animation Cells of a Walking Stick Figure

9. View the videotape, *1988 International Computer Animation Competition Winners*. Afterward, make a list of all the animation sequences you would like to make. Try to make the list as long as you can and try not to list any that you viewed on the tape. Try to be as specific as you can, for example: "a short sequence of a boy playing with a yo-yo" would be preferred over "a cartoon for Saturday T.V."

10. From the above list, select an animation sequence for your next project. Go back and scan through the book, *Animation*, again. Look for ideas for your project. Develop the storyboard. With paper and pencil, make preliminary sketches of each of the cells. At first, try to make it simple. You can elaborate later. Before you begin to work with the computer, check with your instructor, for her/his suggestions.

11. If the equipment is available, record your sequence on videotape. Consider entering your project in an animation competition.

Evaluation Criteria: The student should be given a short quiz where she or he is to define four randomly selected terms from this unit. Her/his definitions should conform to definitions found in the recommended or suggested texts or videotapes. She or he should be able to define three of the four terms successfully.

The student should also be evaluated on how well she or he can describe a computer generated animation application (see Student Activities, Part 7). The student's description of an application should include the advantages of using computer animation over conventional techniques, the types of computers used, careers associated with the application, and some specific examples of work accomplished with this application. Students should be given extra credit if their examples are not like those described in the videotape or the text (required additional research), and if they demonstrate an exceptional amount of enthusiasm in their presentation.

The third and last part of the evaluation should be based on the student's skill and enthusiasm in developing the animation sequences. Special attention should be given to Student Activity, Parts 9 and 10. The animated sequence should contain at least six cells and represent a smooth fluid movement of the image. The storyboard and individual cells will be skillfully drawn (will compare favorably to an example prepared by the instructor). Students should be given extra credit for: character animation, the movement of humans or animals, as opposed to a 2D object sliding across the screen; 3D effects, such as shadows, shading, and highlighting; and added features such as backgrounds, sound, etc.

Microcomputer Graphics Applications
DESKTOP PUBLISHING

Introduction: Desktop publishing systems are interactive computer systems that perform page or document processing. Text and graphics are manipulated on a page or pages to create reports, newsletters, books, pamphlets, and fliers. These systems use personal computers, scanners, and printers (usually laser printers) to perform those tasks necessary to publish a document--create the original material, layout pages, and print the final camera-ready copy. As with many other computer graphics applications, desktop publishing is revolutionizing the communications industry.

Enabling Objective: Before beginning this activity, the student should have completed Activity COG-003, How to Operate the Apple Computer. He or she should be able to turn the computer off and on, be familiar with the keyboard, be able to load and save programs to the disk drive, and be able to use the printer. It would also be desirable to have the student complete Activity COG-011, Computer Graphics Painting, before beginning this activity.

Performance Objective (15-20 days):

TASK/COMPETENCY--COG-014

Upon completion of this activity, the student should be able to describe a publishing system application and to use a system to produce camera ready copy for publication. The student should also be able to recall definitions for the following:
:

text fonts	greeking
justification	leading
binding	letterspacing
photocopying	snap-to
offset printing	kerning
proofreading	optical character readers
WYSIWYG	scanners

CONDITIONS

The above tasks should be performed under the following conditions:

a) The terms should be recalled from memory.

b) The student should be provided resource materials when she or he is asked to write a description of or orally describe an application.

c) To generate the camera ready copy, the student should be provided with rough drafts of articles and news items; an Apple IIe, IIc, or IIgs computer system with at least one disk drive (2 are suggested); appropriate input devices; a color monitor; and a desktop publishing software application package.

STANDARDS OF PERFORMANCE

The above tasks should be performed to the following standards:

a) The student should be expected to provide a definition (similar to that found in the current computer graphics literature) for six out of eight terms selected randomly.

b) The description of an application should include the advantages of using desktop publishing systems over conventional techniques, the types of computers used, careers associated with the application, and some specific examples of work accomplished with this application.

c) The camera ready copy should require the use of at least five of the desktop publishing system functions. The camera ready copy should be free of spelling, hyphenation, and grammatical errors. It should also conform with page layout design principles.

Selected Instructional Materials: An Apple IIe (may need to have ROM enhancement), IIc, or IIgs computer with 64K or 128K of RAM memory and a matrix printer; monitor; disk drive (preferably two); and the desktop publishing application program *Personal Newsletter, Publish It!, Springboard Publisher, Medley,* or *The Newsroom.* Also needed are the manuals that are included with the chosen software. Some of the above software requires 128K of RAM memory and some may be used with laser printers, as well as matrix printers. Matrix printers that are considered near letter quality (LQ) or laser printers are preferred for this activity. Other devices, especially a scanner and a photocopy machine or offset press, would also be very useful.

Teacher References

Recommended

Books:

The user's guide or user's manual that is supplied with the software package or packages chosen for this activity.

McNeil, D. (1987). *Compute!'s quick and easy guide to desktop publishing*. Greensboro, NC: Compute! Publications Inc.

Parker, R.C. (1988). *Looking good in print.*. Chapel Hill, NC: Ventana Press.

Software:

One of the following page oriented document and newsletter processors:

Milliken Publishing Co. (1988). *Medley*. Available form the Milliken Publishing Co., 1100 Research Blvd., St. Louis, MO 63132-0579 or a local computer software dealer.

Softsync, Inc. (1987). *Personal newsletter*. Available from Softsync, Inc., 162 Madison Ave., New York, NY 10016 or a local computer software dealer.

Springboard Software. (1988). *Springboard publisher*. Available from Springboard Software, 7808 Creekridge Circle, Minneapolis, MN 55435 or a local computer software dealer.

Springboard Software. (1985). *The newsroom.*. Available from Springboard Software, 7808 Creekridge Circle, Minneapolis, MN 55435 or a local computer software dealer.

Timeworks. (1988). *Publish it!* Available from Timeworks, 444 Lake Cook Road, Deerfield, IL 60015 or a local computer software dealer.

Suggested Software:

A word processor similar to one of the following or a paint program similar to those used in COG-013 may be used in conjunction with one of the above document processors. Check to see if the document processor you have selected will accept files from the word processor or the paint program, before you make a purchase:

Lissner, R.J. & Apple Computer, Inc. (1983-86). *AppleWorks*. Available from the Claris Corporation, 20525 Mariani Avenue, Cupertino, CA 95014 or a local computer software dealer.

Scholastic, Inc. (1987). *Bank street writer III*. Available from Scholastic, Inc., 730 Broadway, New York, NY 10003 or a local computer software dealer.

StyleWare. (1987,1988). *Multiscribe*. or *Multiscribe GS*. Available from StyleWare, 5250 Gulfton, Suite 2E, Houston, TX 77081 or a local computer software dealer.

Software similar to the following is available for special publishing applications such as designing banners, greeting cards, certificates, etc:

Balsam, D. & Kahn, M. (1984). *The print shop* or *Print shop GS*. Available from Broderbund Software, 17 Paul Drive, San Rafael, CA 94903-2101.

Baudville. (1986). *Award maker plus*. Available from Baudville, 5380 52nd Street S.E., Grand Rapids, MI 49506.

Springboard Software. (1986). *Certificate maker*. Available from Springboard Software, 7808 Creekridge Circle, Minneapolis, MN 55435

Bantam. (1987). *Walt Disney card and party shop*. Available from Bantam, 666 Fifth Avenue, New York, NY 10103.

Some of the desktop publishing software comes with clip art files or makes these files available to you for an additional price. Sometimes compatible files are available through third parties. You may find these to be very helpful.

Software similar to the following can also be useful in this activity if your students need help with writing articles, spelling or typing:

Super Scoop
Helpful Writer
Persuasive Essay
The Sensible Speller
Junior Typer
Typing Tudor IV
Mastertype

The above software is available from several sources; however, one source that has all of them plus other software relative to teaching writing, reading, spelling, and grammar is Opportunities for Learning, Inc., 20417 Nordhoff St., Department V A 3, Chatsworth, CA 91311. Refer to their catalog, *The Computer Approach to English*.

Periodicals: *Publish*. PCW Communications Inc., 501 Second Street, San Francisco, CA 94107. Subscription Information available from Subscriber Services, P.O. Box 55400, Boulder, CO 80322. A monthly publication.

Videotapes:

Teacher Activities

1. Familiarize yourself with desktop publishing by reading *Compute!'s Quick and Easy Guide to Desktop Publishing*. written by McNeil. Chapters 1, 2, 3, 7, 9, and 11 are especially important. Although this publication concentrates on desktop publishing with a Macintosh or IBM PC computer, keep in mind that the principles, concepts, and terms are alike regardless of the computer used. The main difference in desktop publishing on a Macintosh or IBM PC and desktop publishing on an Apple IIe, IIc, or IIgs is the quality of the output. With a laser printer as the output device on an Apple IIgs system and the better IIgs desktop publishing software, there is very little difference. Also read Section 1, and scan the remainder of Parker's *Looking Good in Print*.

2. Familiarize yourself with the software that you plan to use in this activity--desktop publishing, word processor, graphics paint systems, clip art files, and other application programs or files.

4. If time permits, work through the student activities.

5. Prepare to demonstrate the software packages you plan for your students to use.

6. Be prepared to offer suggestions for the student desktop publishing activities. This may include developing cooperative projects with other classes (journalism, English, etc.) student clubs, local civic organizations, or the school administration. Cooperative arrangements could have your students publishing materials for these classes, clubs, organizations, etc. Caution should be taken, however, to prevent the demands and deadlines of these other parties from dictating your curriculum.

Student References

Recommended

Books:

The user's guide or user's manual that is supplied with the software package or packages chosen for this activity.

McNeil, D. (1987). *Compute!'s quick and easy guide to desktop publishing*. Greensboro, NC: Compute! Publications Inc.

Parker, R.C. (1988). *Looking good in print*. Chapel Hill, NC: Ventana Press.

Software:

One of the following page oriented document and newsletter processors:

Milliken Publishing Co. (1988). *Medley*.

Softsync, Inc. (1987). *Personal newsletter*.

Springboard Software. (1988). *Springboard publisher*.

Springboard Software. (1985). *The newsroom*..

Timeworks. (1988). *Publish it!*

Suggested

Software:

Software similar to the following is available for special publishing applications such as designing banners, greeting cards, certificates, etc:

Balsam, D. & Kahn, M. (1984). *The print shop or Print shop GS*.

Baudville. (1986). *Award maker plus*.

Springboard Software. (1986). *Certificate maker.*

Bantam. (1987). *Walt Disney card and party shop.*

A word processor similar to one of the following or a paint program similar to those used in COG-013 may be used in conjunction with one of the above document processors.

Lissner, R.J. & Apple Computer, Inc. (1983,1986). *AppleWorks.*

Broderbund Software, Inc. (1987). *Bank street writer III.*

StyleWare. (1987,1988). *Multiscribe. or Multiscribe GS.*

Some of the desktop publishing software comes with clip art files or makes these files available to you for an additional price. Sometimes compatible files are available through third parties. You may find these to be very helpful.

Software similar to the following can also be useful in this activity:

Super Scoop
Helpful Writer
Persuasive Essay
The Sensible Speller
Junior Typer
Typing Tudor IV
Mastertype

Periodicals:

Publish. PCW Communications Inc., 501 Second Street, San Francisco, CA 94107.

Student Activities

1. Read Part 1 of and scan other parts of *Compute!'s Quick and Easy Guide to Desktop Publishing*. written by McNeil and Section 1 of Parker's *Looking Good in Print*. As you read, make a list of new terms that are associated with desktop publishing. When you are through, you should be able to define the following terms. If you don"t understand some of them, discuss them with your teacher. Rather than do all of this reading at one time, mix it in with some of the subsequent activities

text fonts	greeking
justification	leading
binding	letterspacing
photocopying	snap-to
offset printing	kerning
proofreading	optical character readers
WYSIWYG	scanners

2. Observe your instructor as she or he demonstrates the desktop application software that has been selected for you to use in this activity.

3. Practice using the desktop publishing application program your instructor has selected for this activity:

   a) Read the user's guide that accompanies the software package.

   b) Use the software package to design one of the following:

   (1) a greeting card.
   (2) a poster for the bulletin board announcing a:
   club meeting,
   a ball game,
   special event such as a cake sale, or
   an item for sale.
   (3) a business card.
   (4) letterhead stationery.
   (5) a schedule for a ball club.
   (6) a poster for a candidate in a class or student council election.
   (7) the school calendar for next month.
   (8) a teacher's daily schedule.

   Try to be original in your design. Remember, you can incorporate both text and graphics.

4. Visit a local print shop, newspaper, industry, or office that uses computers to publish materials. While on your visit, determine what software and hardware they use, the training required of the people involved in the publication, and the types of publications they generate on their system. It might also be interesting to find out if and how they published these materials before they acquired the desktop publishing system. Has the system improved productivity?

5. With your instructor's help, organize your class or group into a simulated publication organization and publish a newsletter, newspaper, brochure, or other publication. Your organization may need to have a publisher, editor, artists, writers, reporters, proof readers, press operators, and computer operators. More than likely, you and your classmates will want to perform one or more of these jobs by rotating through them.

Evaluation Criteria: The student should be given a short quiz where she or he is to define eight randomly selected terms from this unit. Her/his definitions should conform to definitions found in the recommended text. She or he should be able to define six of the eight terms successfully.

The student should also be evaluated on how well she or he can describe a computer desktop publishing application. The student's description of an application should include the advantages of using desktop publishing over conventional techniques, the types of computers used, careers associated with the application, and some specific examples of work accomplished with this application.

The third and last part of the evaluation should be based on the student's skill and enthusiasm in developing the publications. Special attention should be given to Student Activity, Parts 3 and 4. The publications should have required camera ready copy that used at least five desktop publishing system functions to produce. The publications should also be free of spelling, hyphenation, and grammatical errors. They should also conform with page layout design principles.

COMPUTER GRAPHICS RESOURCE LIST

Books

Allan, J. J. (1979). *The CAD/CAM glossary*. Dallas, TX: Productivity International, Inc.

A comprehensive glossary of computer-aided-design/computer-aided-manufacturing terms.

Apple Computer, Inc. (1983). *Apple II DOS user's manual*. Cupertino, CA: Author.

An Apple reference manual describing the Apple DOS 3.3 operating system. It covers those commands which permit the user to do tasks which call for the use of the disk drives.

Apple Computer, Inc. (1983). *Apple II ProDOS user's manual*. Cupertino, CA: Author.

An Apple reference manual detailing the Apple ProDOS operating system. It explains how to use the ProDOS User's Disk to manage information on disks.

Apple Computer, Inc. (1983). *Applesoft BASIC programmer's reference manual*. Reading, MA: Addison-Wesley.

A reference manual for the Applesoft BASIC programming language as used by the Apple II series of computers. It is for use by those that have some programming experience using the BASIC language.

Apple Computer, Inc. (1985). *The Applesoft tutorial*. Reading, MA: Addison-Wesley.

Apple Computer, Inc. (1986). *A touch of Applesoft BASIC*. Cupertino, CA: Author.

These manuals are for the beginning programmer who plans to use the Applesoft BASIC language to program the Apple II series of computers.

Blair, P. (1986). *Animation*. Tustin, CA: Foster Art Service Inc.

A very good how-to-do-it book on animation. It is well illustrated with Disney style characters.

Bowman, D.J. (1984). *The CAD/CAM Primer*. Indianapolis, IN: Howard W. Sams & Co., Inc.

An excellent source of information concerning CAD/CAM (more CAD than CAM) concepts, hardware, software, and terminology. It contains a very good glossary of CAD/CAM terms. This is a generic textbook not a specific CAD application textbook.

Apple Computer, Inc. (1987). *BASIC programming with ProDOS*. Reading, MA: Addison-Wesley.

Explains the ProDOS commands and shows how to incorporate them into Applesoft BASIC programs--a must for the BASIC programmer who is using the ProDos operating system.

Benton, R. & Balcer, M.S. (1988). *The official Print Shop handbook*. New York, NY: Bantam Books.

A book that provides ideas, tips, and designs for users of the *Print Shop* application software.

Berryman, G. (1979). *Notes on graphic design and visual communication*. Los Altos, CA: William Kaufmann, Inc.

Explains graphic design concepts with visual examples. An easy to read book covering designing with marks, symbols, logos, pictographs, and type. The stages of the graphic design process are also very well explained. An excellent resource book for many of the activities found in this curriculum guide, especially the one on desktop publishing.

Bertoline, G.R. (1985). *Fundamentals of CAD*. Albany, NY: Delmar Publishing, Co.

A comprehensive and generic CAD text. It has an excellent introduction to CAD systems in the first section. The second section describes a typical CAD system, and the third section contains instruction relative to creating a drawing using a CAD system-including appropriate drawing exercises. The last section contains a discussion of CAD applications. Each chapter has an excellent glossary. A good introductory text for grades 9 - 12.

Bitter, G., & Severe, S. (1986). *Programming in BASIC for the Apple II series*. Reading, MA: Addison-Wesley.

A very good BASIC text for beginners. It covers the basics without the clutter of those details associated with infrequently used commands.

Consumer Guide Editors. (1983). *Illustrated computer dictionary*. Skokie, IL: Publications International.

A good computer dictionary that is also well illustrated.

Fox, D., & Waite, M. (1984). *Computer animation primer*. New York: McGraw-Hill.

This book is in two parts. The first part is a good overview of computer animation for the more advanced student (the reading level and mathematics are advanced). The second part deals with animation on an Atari computer. Although the exercises are for the Atari, there is much useful information for someone interested in animation, but using a different computer.

Goetsch, D.L. (1983). *The CAD/CAM workbook*. Cincinnati, OH: South-Western Publishing Co.

Designed to be used with a traditional drafting text, this workbook can be used in a setting where there are no computers (according to the author). It exposes the student to the terminology and concepts of CAD/CAM. Each unit is designed as a self instructional module.

Goetsch, D.L. (1985). *Computer-aided drafting*. Englewood Cliffs, NJ: Prentice-Hall, Inc.

A follow-up text to *Introduction to Computer-Aided Design* by Goetsch. This text covers the subject better than Goetsch's introductory text. It has more breadth and depth and it leaves out some of the unnecessary information found in the earlier text. As such, it is a very good text for the CAD activity in this curriculum guide.

Hubbard, S. W. (1983). *The computer graphics glossary*. Phoenix, AZ: Oryx Press.

A comprehensive glossary on CAD/CAM. Contains terms unique to computer graphics as well as some of the more common CAD/CAM product names.

Jankel, A., & Morton, R. (1984). *Creative computer graphics*. Cambridge, England: Cambridge University Press.

An excellent publication presenting computer graphics from the pioneering efforts of the 50's to the current achievements of modern exponents from several countries. The book explores computer graphic images through the techniques and technology used to create them. It is well illustrated.

Jones, A.B. (1982). *I speak BASIC to my Apple*. Rochelle Park, NJ: Hayden Book Co.

A very good computer literacy book for the beginning computer student. It teaches concepts plus the BASIC language.

Kemnitzer, R.B. (1983). *Rendering with markers*. New York: Watson-Guptill Publications.

An excellent resource for students learning to illustrate with color. Although it explains how to illustrate with markers, the same principles can be applied to computer paint systems.

Lamit, L. G., & Paige, V. (1987). *Computer-aided design and drafting*. Columbus, OH: Merrill.

A well-written generic CADD textbook. It does a very good job of explaining the key concepts of a computer aided design system. Suitable for the more advanced student.

Laybourne, K. (1979). *The animation book*. New York: Crown.

A comprehensive text describing all types of animation and animation tools, techniques, and resources. The beginning chapters include a very good introduction to computer animation. The remaining portions of the book are also recommended for teachers and those students with an advanced interest in animation.

Martin, C.L. (1968). *Design graphics*. New York, NY: The Macmillan Company.

Although it has an old publication date, this is still an excellent reference for the teacher who is planning a presentation on illustration and rendering techniques in anticipation of students working with a paint or animation application program.

McNeil, D. (1987). *Compute!'s quick and easy guide to desktop publishing*. Greensboro, NC: Compute! Publications Inc.

An overview of the revolution known as desktop publishing. Includes a history of the field of printing, the terms used in printing, writing copy, designing the page, selecting type and graphics, and organization and dissemination of the final document. May not be suitable for students who do not read well. However, it is recommended as a reference for those using the desktop publishing activity.

Myers, R.E. (1984). *Microcomputer graphics for the Apple computer*. Reading, MA: Addison-Wesley.

An excellent source for the advanced Apple II graphics student. Contains explanations of most of the advanced techniques used by professional Apple graphics programmers. It can be purchased with a diskette containing those programs listed in the text.

McKissick, M. L. (1987). *Computer-aided drafting and design*. Englewood Cliffs, NJ: Prentice-Hall.

A generic CADD textbook for the more advanced student.

Minnesota School Districts Data Processing Joint Board. (1983). *Apple II microcomputer quick reference guide*. Minneapolis: The author.

A very handy fan-fold reference guide designed for students by teachers. For use with the Apple II, II+, and IIe. It covers some of the basic computer operation procedures.

Parker, R.C. (1988). *Looking good in print.* Chapel Hill, NC: Ventana Press.

An excellent generic reference for desktop publishing design. It emphasizes page design and can be used with any desktop application program. Includes sections on the elements of graphic design, illustrations of the elements through page makeovers, and a detailed discussion related to the different products of desktop publishing--newsletters, newspaper and magazine ads, brochures, business correspondence, etc.

The Public Domain Exchange. (1987). *The best of Apple public domain software, edition II.* Santa Clara, CA: the author.

A comprehensive listing of non-copyrighted public domain software, copyrighted public domain software, and shareware for the Apple II--software which is free or inexpensive. The software is catalogued by application. Of particular interest to teachers and students using this curriculum guide are the listings under art, CAD, and utilities.

Reid-Green, K. (1984). *Games computers play.* Bedford, MS: Digital Press.

A very good reference for the student who is a good programmer and wants to develop computer games with computer graphics. Computer code used in the book is structured and machine independent.

Sayre, C. L. (1986). *Design and graphics using the Apple computer.* Reston, VA: Reston Publishing Co.

A very good reference for those who want an in-depth understanding of graphics programming using Applesoft BASIC. Those choosing to use this book should be familiar with the BASIC language and the Applesoft graphic commands.

Schmieman, S. (1985). *MacPaint drawing, drafting, design.* Bowie, MD: Brady Communications Company, Inc.

This book may be of use for those teachers and students using some of the new Apple IIgs paint software that uses the *MacPaint* format. In addition to specific instructions in how to use the *MacPaint* functions, it has good ideas for painting activities and exercises.

Seymour, R.D., Ritz, J. M., & Cloghessy, F.A. (1987). *Exploring communications.* South Holland, IL: Goodheart-Willcox Co.

A very good resource for those studying the technological field of communications.

Sordillo, D. A. (1983). *The personal computer BASIC(S) reference manual.* Englewood Cliffs, NJ: Prentice-Hall.

A handy reference for those programmers who are interested in rewriting a computer program for a different personal computer. It covers and compares the different dialects of the BASIC language used by the IBM PC, Apple II, Atari, TRS Color, TRS Mod III, and the Commodore VIC-20 computers.

Spencer, D. D. (1986). *The illustrated computer dictionary (3rd ed.).* Columbus, OH: Merrill.

A comprehensive and well illustrated dictionary of computer terminology.

Ungar, J. (1985). *Rendering in mixed media.* New York, NY: Watson-Guptill Publications.

A good resource for the teacher who is preparing presentations on rendering for his or her students. Although the text deals with mixed media, many of the rendering principles apply to a computer paint system, which simulates many different media.

Voisinet, D.D. (1987). *Computer-aided drafting and design, concepts and applications.* New York, NY: Gregg Division of McGraw-Hill.

A generic and more advanced CAD textbook suitable for teachers and more advanced students. In addition to covering those topics found in Voisinet's introductory text, it covers some more advanced topics such as 3-D modeling, partial programming, the effects of color, and CAD/CAM.

Voisinet, D.D. (1986). *Introduction to computer-aided drafting (2nd edition).* New York, NY: Gregg Division of McGraw-Hill.

A generic introductory CAD textbook suitable for those 9 - 12 graders who are reading at their expected level. The text covers the topics CAD uses, equipment, and methods of use. It also covers methods of creating drawings, the typical CAD functions, and the CAD professional's tasks.

Williams, K., Kernaghan, B., & Kernaghan, L. (1983). *Apple II computer graphics.* Bowie, MD: Robert J. Brady Co.

A very good book for those who wish to do computer game type animation using Applesoft BASIC. The reader should be very familiar with Applesoft BASIC and the Apple operating system before trying to read. The reader should also understand hexadecimal and binary numbering systems--the first chapter deals with these topics.

Winkler, C. (1983). *The computer careers handbook.* New York: Arco Publishing Co.

This book provides an overview of the computer field, the jobs available, salaries paid, and the training and education needed. As with many computer career information books, it is slightly dated.

Periodicals

A+ Ziff Davis Publishing Company, One Park Avenue, New York, NY 10016. Subscriptions should be sent to *A+*, P.O. Box 52324, Boulder Colorado 80321-2324. A monthly publication.

Information, tutorials, product reviews, and industry news about the Apple II product line. Suitable for grade levels, 9 - 12.

Cadence. Cadence Publications, 12611 Research Blvd., Suite 367, Austin, TX 78759-9990. A monthly publication.

A magazine for the *AutoCad* user. Contains articles on *AutoCad* applications and helpful techniques, and information about compatible hardware and software. Suitable reading for teachers and students who are very familiar with *AutoCad* software.

Classroom Computer Learning. Peter Li, Inc., 2541 E. River Road, Dayton, Ohio 45439-9988. Published monthly except June, July, August, and December.

Articles, product news, shoppers guide, and answers to commonly asked questions about computers and their use in the elementary and secondary classroom. An excellent publication for teachers who use computers in their classrooms.

Computer Graphics Today. Media Horizons Inc., 50 West 23rd Street, New York, NY 10010. Subscriptions should be sent to *Computer Graphics Today,* POB 656, Holmes, PA 19043. A monthly publication.

The official publication of the National Computer Graphics Association. Subscription is free to members of the association. Contains announcements about computer graphics products, publications, and events. Suitable reading for teachers and most high school students who have a high interest in the field.

Computer Graphics World. PennWell Publishing Company, P.O. Box 1112, 119 Russell Street, Littleton, MA 01460. Subscriptions should be sent to *Computer Graphics World*, P.O. Box 112, Tulsa, OK 74101. A monthly publication.

Industry news, product announcements, and product reviews about computer graphics applications. A good general interest publication designed for the user. Suitable for teachers and junior and senior level students.

Computer Pictures Magazine. Electronic Pictures Corporation, 2 Village Square West, Clifton, NJ 07011. Published bimonthly.

A technical magazine with news and trends in the computer graphics field. Emphasis is placed on computer visual art and design, and animation for film and video. Suitable reading for teachers and advanced students.

IEEE Computer Graphics and Applications. A publication of the IEEE Computer Society Executive Committee, The Institute of Electrical and Electronics Engineers, Inc., 1730 Massachusetts Ave., NW, Washington, DC 20036-0101. Subscriptions should be sent to IEEE Computer Society, 10662 Los Vaqueros Circle, Los Alamitos, Ca 90720. A monthly publication.

A very technical publication usually containing articles on computer graphics algorithms and mathematical techniques. It also contains industry news and new product announcements. May be of interest to those senior high school students with an advanced mathematical background.

InCider. CW Communications Peterborough, Inc., Route 101 and Elm Street, Peterborough, NH 03458. Subscriptions should be sent to Subscription Department, P.O. Box 911, Farmington, NY 11737.

An excellent periodical for the Apple II computer family user. It is highly recommended for both students and teachers. It does a very good job of rating software. It also has well written articles that are timely and helpful both for the programmer and the application software user.

Publish. PCW Communications Inc., 501 Second Street, San Francisco, CA 94107. Subscription Information available from Subscriber Services, P.O. Box 55400, Boulder, CO 80322. A monthly publication.

A desktop publishing periodical with a hands-on approach. Includes feature articles dealing with user's tips, applications, and profiles. It also includes reviews of desktop publishing software. Suitable for anyone interested in desktop publishing.

Software

Barbatano, M., Darooge, M., McCullough, E. & Walker, D. (1986). *Take 1 and Take 1 programmer's toolkit*. Available from Baudville, 1001 Medical Park Drive S.E., Grand Rapids, MI 49506 or a local computer software dealer.

A top rated standard hi-res animation application program. It has outstanding features and is easy to use. Allows the user to create "actors", backgrounds, and sounds to create scenes. Scenes can be combined and made into movies with fade-ins, fade-outs, and sound effects. It is reasonably priced and will run on an Apple II plus, IIe, IIc, or IIgs with 64K.

Darooge, M., McCullough, E. & Walker, D. (1984,1988). *Blazing paddles*. Available from Baudville, 1001 Medical Park Drive S.E., Grand Rapids, MI 49506 or a local computer software dealer.

A top rated standard hi-res paint application program for an Apple II plus, IIe, IIc, or IIgs with 64K of RAM. It is reasonably priced, works with a variety of input devices, and is very easy to use.

Langer, V. (1985). *MATC-CAD* Available from the Milwaukee Area Technical College, 1015 North Sixth Street, Milwaukee, Wisconsin 53203.

An excellent application software package for CAD education. The software was developed under a grant from the National Science Foundation specifically for training purposes. It simulates Computervision's *CADDs* software which is very popular in industry. Comes with excellent teaching materials. There are 64K and 128K Apple versions and an IBM PC version.

Lissner, R.J. & Apple Computer, Inc. (1983-86). *AppleWorks, V.2.0*. Available from the Claris Corporation, 20525 Mariani Avenue, Cupertino, CA 95014 or a local computer software dealer.

An integrated application program containing a word processor, a data base manager, and a spreadsheet. Teachers may find the word processor very useful with several of the activities in this curriculum guide, especially the desktop publishing activity. Students may also find the data base manager or the word processor useful in keeping a record of their computer graphic terms. It is also an excellent tool for classroom management. It can be used for laboratory inventory, test files and test making, and as a computerized gradebook. Requires an Apple IIe, IIc, or IIgs with 128K of RAM.

Milliken Publishing Co. (1988). *Medley*. Available from the Milliken Publishing Co., 1100 Research Blvd., St. Louis, MO 63132-0579 or a local computer software dealer.

A desktop publishing application software package that uses Macintosh-like menus and windows. It includes word processing with an 80,000 word thesaurus and spell checker, graphics, and page layout programs all-in-one. Comes with clip art and a back-up disk. Requires an Apple IIgs with 512K of RAM memory.

Mullen, Fred L. (1983). *Autograph*. Available from the Public Domain Exchange, 2074C Walsh Avenue, Dept 20, Santa Clara, CA 95050. Order disks 310,311, & 312.

Inexpensive CAD software. This is public domain software that can be obtained from the Public Domain Exchange for use on the Apple IIe. It is not very user-friendly.

Ohmert, S. (1986). *Animate*. Available from Broderbund Software, Inc., P.O. Box 12947, San Rafael, CA 94913-2947 or a local computer software dealer.

A top rated double hi-res animation application program. The program has the necessary painting tools to create and edit animated cells and background scenes. Scenes can also be drawn with other software such as *Dazzle Draw* and then used with *Animate*. It uses pull down menus and windows similar to those found in Macintosh applications. The software comes with good documentation and a well illustrated book on animation. It will run on an Apple IIe, IIc, or IIgs with 128K.

Scholastic, Inc. (1987). *Bank Street writer III*. Available from Scholastic, Inc., 730 Broadway, New York, NY 10003 or a local computer software dealer.

Considered to be the best word processor for home and school use, this software is extremely easy to learn and use. It contains a spell checker and a thesaurus. Requires any Apple II computer with 128K of RAM.

Silva, D. (1988). *Deluxepaint II*. Available from Electronic Arts, 1820 Gateway Drive, San Mateo, CA 94404 or a local computer software dealer.

A top rated super hi-res graphics paint application program. This program has all of the functions normally found in a paint program plus some not expected in an Apple application. It will run on an Apple IIgs with 768K of RAM.

Snider, D. (1984,1987). *Dazzle draw*. Available from Broderbund Software, Inc., P.O. Box 12947, San Rafael, CA 94913-2947 or a local computer software dealer.

A top rated double hi-res paint application program. This is an excellent program with which to introduce the user to the capabilities of a computer graphics 2D paint application. Contains most of the functions found on a much more expensive system. Will run on an Apple IIe, IIc, or IIgs with 128K.

Softsync, Inc. (1987). *Personal newsletter*. Available from Softsync, Inc., 162 Madison Ave., New York, NY 10016 or a local computer software dealer.

This desktop publishing software will allow the user to create many kinds of documents, but it is especially suited to creating newsletters. The program has graphics functions for drawing borders and fill patterns, it will permit from one to ten columns for text layout, has six fonts in ten type styles, and comes with clip art. You may also import graphics from *Print Shop, The Newsroom, Dazzle Draw*, or a Thunderscan (digitizing) unit. It includes the drawing tools to draw your own or edit imported drawings. Requires an Apple IIe (with ROM enhancement), IIc, or IIgs with 128K and an 80 column board.

Springboard Software. (1988). *Springboard publisher*. Available from Springboard Software, 7808 Creekridge Circle, Minneapolis, MN 55435 or a local computer software dealer.

A desktop publishing application that emulates the well known *Pagemaker* application on the Macintosh. This software has three working modes--page processing, word processing, and graphics. It was designed with school and home applications in mind. The clip art that comes with it called, "Works of Art", contains graphics related to educational and holiday themes. Additional clip art may be purchased. Files may be imported from *Mouse Paint, Springboard Art*, and ASCII text files.

Springboard Software. (1985). *The newsroom*. Available from Springboard Software, 7808 Creekridge Circle, Minneapolis, MN 55435 or a local computer software dealer.

This is a software classic. It contains word processing, graphics, banner, and telecommunications functions. Although limited when compared to some of the newer page and newsletter processing software, it is still an effective educational tool. A lot of clip art is available for use with it and it comes with comprehensive documentation including suggestions for gathering information, writing stories, and organizing a classroom newspaper. Requires an Apple II Plus, IIe, or IIc with 64K and one disk drive (two are suggested). Works with any number of matrix printers. Also available for the IBM PC.

StyleWare. (1987). *Multiscribe V.3.3*. or *Multiscribe GS V.3.0*. Available from StyleWare, 5250 Gulfton, Suite 2E, Houston, TX 77081 or a local computer software dealer.

A full featured word processing program that uses double hi-res graphics and proportional spacing to emulate the *MacWrite* word processor used on the Macintosh computer. It uses the pull down menus with a mouse or the keyboard and has a multitude of fancy fonts. The GS version is especially nice with its spell checker and thesaurus, graphics capabilities, and the ability to directly import *AppleWorks* files. *Multiscribe 3.3* requires an Apple IIe, IIc, or IIgs with 128K and an 80 column card. *Multiscribe GS 3.0* requires a IIgs with 512K.

Timeworks. (1988). *Publish it!* Available from Timeworks, 444 Lake Cook Road, Deerfield, IL 60015 or a local computer software dealer.

This desktop publishing software has a Macintosh type interface with pull-down windows, icons, etc. and a full featured word processor. It will allow the user to import files from *Dazzle Draw, Mouse Paint, Beagle Graphics, AppleWorks, Bank Street Writer*, and ASCII files from other word processors. It can be used with a mouse or a joystick.

Versacad Corporation. (1987). *Entry level CADApple*. Available from Versacad Corporation, 7372 Prince Drive, Huntington Beach, CA 92647 or a local computer software dealer.

An entry level CAD software application package that uses a subset of the functions available in the popular *CADApple* software. Comes with a textbook and an audio tape.

Version Soft, Inc. (1986). *Paintworks plus*. Available from Activision, Inc., P.O. Box 7287, Mountain View, CA 94039 or a local Apple computer dealer.

A paint application program and an animation application program all in one. The paint program has the same painting tools as MacPaint and it uses the pull down menus and windows found in Macintosh programs. It has the ability to use 16 colors in one painting selected from a palette of 4096 colors. Easy to use and comes with good documentation. Can be used on an Apple IIgs with 512K.

Videotapes

CIM Solutions. (1986). *Computer aided design: a video evaluative report I*. Available from CIM Solutions, University Station, P.O. Box 7041, Provo Utah 84602.

CIM Solutions. (1986). *Computer aided design: a video evaluative report II*. Available from CIM Solutions, University Station, P.O. Box 7041, Provo Utah 84602.

These two videotapes evaluate a variety of popular computer aided design systems by having the viewer look over the shoulder of a CAD professional as he/she draws a benchmark object using each of the different systems. A good idea; however, the professionals are not always very well trained on the system they are using and some of them make some serious drafting errors. The tapes do allow the viewer to see the different functions supplied by systems.

DCA Educational Products, Inc. (1982). *Apple IIe, VHS*. from the *Computer operations series*. Bedminster, Pennsylvania 18910: The author.

An older video tape that does an excellent job of introducing a new user to the Apple family of computers. It demostrates the basic tasks of turning the computer on, using the disk drive, and running programs. It may not be as appropriate for students who are using the Apple IIgs as it is for those using the Apple IIe or IIc.

Dupont, C. & Cinemagic Productions. (1987). *Computer magic*. Available from Cinemagic Productions, 537 Jones Street, #898 San Francisco, CA. 94102.

An excellent presentation by one of the pioneers of computer animation, Mr. Robert Abel. He takes the viewer behind the scenes to describe the production of some of the more popular computer animated T.V. commercials.

Electronic Pictures Corporation. (1987). *Computer graphic special on TV commercials and animation*. Available from Frost and Sullivan, Inc., Department R-1, 106 Fulton Street, New York, NY 10038.

An exciting anthology of some of the best computer animation ever produced, presented with fascinating explanations of many of the creative processes and production techniques actually used.

Electronic Pictures Corporation. (1983). *Creative computer graphics*. Available from Frost and Sullivan, Inc., Department R-1, 106 Fulton Street, New York, NY 10038.

An excellent introduction to computer graphics. The video surveys the different applications of computer graphics, defines some key terminology, and explains some of the key technological advances in the field.

Frost and Sullivan, Inc. (1986). *SIGGRAPH '86*. A Frost and Sullivan video report. Available from Frost and Sullivan, Department R-1, 106 Fulton Street, New York, NY 10038.

A guided video tour of the SIGGRAPH '86 Exhibition. SIGGRAPH is the Special Interest Group on Graphics of the Association of Computing Machinery (ACM). The tour features brief presentations of more than 40 computer graphics products and in-depth presentations of about 20 more products.

Frost and Sullivan, Inc. (1987). *Workstations/video text report (VTR) 001*. A Frost and Sullivan video text report. Available from Frost and Sullivan, Department R-1, 106 Fulton Street, New York, NY 10038.

A video on state-of-the-art computer graphics hardware in the workstation category. Also contains a verbatim transcript of the narration and commentary. This is a very technical video and should only be shown to students after a discussion of some key computer graphics hardware terminology. With the presentation of each type of workstation there are examples of the output that can be created on the workstation. This makes it a very interesting video for anyone interested in computer graphics.

Frost and Sullivan, Inc. (1987). *Graphic arts systems/video text report (VTR) 002*. A Frost and Sullivan video text report. Available from Frost and Sullivan, Department R-1, 106 Fulton Street, New York, NY 10038.

A video on state-of-the-art computer graphics 2D paint and 3D modeling systems. Also contains a verbatim transcript of the narration and commentary. This is a very technical video and should only be shown to students after a discussion of some key computer graphics terminology related to paint and modeling systems. With the presentation of each type of system, there are examples of the output that can be created on the system. This makes it a very good video for anyone interested in computer graphics.

Herr, L., & Zaritsky, R. (1988). *Visualization: state of the art*. A special issue of the ACM/SIGGRAPH Video Review--Volume 30. Association for Computing Machinery, 11 West 42nd Street, New York, NY 10031.

A comprehensive coverage of state-of-the-art computer graphics. This video is over two hours in length and covers historical events (Sketchpad at MIT, 1964), new developments and trends in hardware, developments in graphics programming and rendering, computer graphics systems, a special look at computer graphics in scientific computing and medical imaging, and a look at the future of computer graphics. The video features most of the best known computer graphic scientists as speakers and plenty of computer generated graphics. An excellent video; it should be viewed by all computer graphics enthusiasts.

Note: The teacher should allow plenty of time to review the video. There are some segments that are only appropriate for the most advanced students. Therefore, he or she should carefully select those segments that are appropriate for his or her students. The video comes with a well documented script and topical outline which will help in the review.

Massachusetts Institute of Technology. (1987). *MIT media laboratory*. Cambridge, MA: Available from the author.

A videotape with segments on each of the major research projects at MIT's media laboratory where some of the most exciting computer application research is taking place. Much of the research utilizes computer graphics.

National Computer Graphics Association. (1987). *1987 International computer animation competition winners*. Available from NCGA, 2722 Merrilee Drive, Fairfax, VA 22031.

National Computer Graphics Association. (1988). *1988 International computer animation competition winners*. Available from NCGA, 2722 Merrilee Drive, Fairfax, VA 22031.

Videotapes of most of the 1987 and 1988 International Computer Animation Competition winners. This competition is sponsored each year by the National Computer Graphics Association and each year the NCGA releases a videotape of most of the animation sequences for which awards were presented. Some winners are not included because they were originally developed for customers who do not wish to release them.

Ocean Communications, Inc. (1987,1988). *On becoming a desktop publisher*. Ocean Communications, Inc., 1641 N. First St., Suite 160, San Jose, CA 95112.

Includes two detailed videotapes. One is a tutorial including a discussion of computers, layout, artwork, printers, fonts, large screen displays, and scanners used in desktop publishing. The second is a demonstration tape where the better known page layout and artwork software packages and the three Apple laser printers are presented. Even though these tapes are prepared for business use of desktop publishing with the Macintosh and IBM PC compatibles, they can give the student a very good overview of the field.

SIGGRAPH/ACM. *SIGGRAPH video review--issues 1 through 24*. Association for Computing Machinery, 11 West 42nd Street, New York, NY 10031.

These issues illustrate the latest concepts in computer graphics and interactive techniques. Begun in 1980 with issue 1, they have been made each year since. Available on 3/4" U-matic and VHS.

Silicon Graphics Computer Systems. (1987). *Superworkstation demonstration*. Silicon Graphics, Corporate Headquarters, 2011 Suerlin Road, Mountain View, CA 94043.

A promotional video on Silicon Graphics Workstations. It surveys very well the current applications related to CAD and CAE. An excellent video for students interested in state-of-the-art mechanical engineering applications.

GLOSSARY

absolute coordinates.

The value of the X,Y, and Z Cartesian coordinates with reference to the origin of the coordinate system.

AEC.

An abbreviation for Architectural, Engineering, and Construction. A segment of industry that uses a special type of computer aided design software for architectural design, facilities planning and management, highway design, and many other architectural or construction related applications.

algorithm.

A step by step procedure for solving a problem. Often used in computer programming to describe a procedure with a finite number of well defined steps for solving a problem within a program.

aliasing.

The stair stepping effect that is caused by drawing a non-horizontal or non-vertical line on a raster refresh display device or any other raster type of device. The poorer the resolution of the device the more pronounced the aliasing. Sometimes referred to as the "jaggies."

alphanumeric.

Refers to all of the characters in a standard alphabet, numerals 0-9, the control characters, mathematical symbols, punctuation marks, and other characters and symbols found on a conventional computer keyboard.

anti-aliasing.

A process where the visual effects of aliasing are reduced.

application software.

A set of computer programs design to solve problems or meet the requirements of a specific job or industry. In computer graphics, such software may be designed specifically for designing machine devices (CAD), document processing (Desktop Publishing), rendering a 2D image (Paint System), architectural drawing (AEC), etc.

array.

A series of terms, values, or elements arranged in a meaningful way. In a computer program, it refers to an area or file where data is stored. Also referred to as a matrix.

aspect ratio.

The relationship between height and width. In computer graphics it can be used to describe the viewport of a display device, the window of the graphic data base, text, or any graphic element. RATIO = HEIGHT/WIDTH or RATIO = WIDTH/HEIGHT.

assembly language.

A low level computer language. Consists of mnemonic statements that have a one to one correspondence with specific machine language instructions of the computer.

B-spline.

Refers to a simulated curve or curved surface generated by a graphics program. The generated curve or curved surface consists of a large number of minute vectors calculated to pass through a number of given points.

BASIC.

An acronym which stands for Beginner's All-purpose Symbolic Instruction Code. Originally developed at Dartmouth College, it is one of the more commonly used high level programming languages for personal computers.

batch processing.

Refers to a method of processing computer jobs, wherein the jobs are processed consecutively and without human interference. The opposite of interactive.

binary.

A characteristic where only two alternatives exist. Digital computers use a binary numbering system. This is a numbering system where the base is 2, and there are only two counting numbers, a 0 and a 1.

binding.

The act or method used to fasten pages of a document together usually with a document cover. Common methods of binding are saddle stitching, spiral binding, perfect binding, and edition binding.

bit map.

That portion of the computer's memory devoted to describing the image being displayed on the display device. With a raster refresh display device, there is a one to one correspondence between a bit or a group of bits in the frame buffer or bit map and a pixel on the display device screen.

bit.

A contraction of "binary digit." A single character in a binary number, either a 0 or a 1. The smallest unit of information in a digital computer.

byte.

A group of binary digits or bits. Sometimes referred to as a word. Usually consists of eight bits.

CAD.

An acronym for Computer Aided Design. A computer system application that is used for designing and drafting products.

CAD/CAM.

An acronym that stands for computer aided design/computer aided manufacture. Refers to an integrated computer system that is capable of controlling the design and the manufacture of a product from one data base.

CADD.

An acronym for Computer Aided Design and Drafting. Some prefer this acronym to CAD when describing a computer system application that is used for designing and drafting products.

Cartesian coordinates.

A system of defining and locating points in 2D and 3D space named for the mathematician, Rene Descartes. Using this system, a point is defined by its location from the origin relative to a set of mutually perpendicular axes, X,Y, and Z. This system is used in all computer graphics applications for locating graphic elements.

CIM.

An acronym that stands for Computer Integrated Manufacture. A computer system where not only is the manufacture and design of a product under computer control (CAD/CAM); but also inventory control, purchasing, inventory and production control, quality assurance, etc.

clipping.

That function performed by a computer graphics program when it discards graphic data that fall outside a specified boundary (the window). In some cases, this may necessitate that new endpoints be computed for the boundary intersections. If this task were not performed, wraparound could occur on the display device screen or an error message may halt execution. Wraparound is when lines are drawn on the display device screen in an uncontrolled manner.

compiler.

Computer software that translates a high level language such as FORTRAN or BASIC into machine language that the computer understands. After the program has been compiled, the machine language compilation, referred to as object code, is executed.

computer animation.

The use of a computer to generate video, film, or computer displayed real time sequences. Animation is the presentation of many still images in rapid succession so as to create the illusion of smooth fluid motion.

computer graphics system.

Refers to all of the software, hardware, and documentation necessary for a computer graphics application.

computer graphics.

That branch of computer technology that concentrates on the creation of graphic images for many different applications to include computer-aided design, desktop publishing, presentation graphics, computer-aided animation, medical graphics, and image processing.

CPU.

An abbreviation for Central Processing Unit. That portion of a computer that contains all of the hardware that controls and executes the programming instructions--the brains of a computer. In a microcomputer, the CPU may be on a single chip called the microprocessor.

CRT.

An abbreviation for Cathode Ray Tube. A vacuum tube similar to the one used in a T.V. It is the technology used in most computer graphics display devices. An electron beam draws images by focusing on a portion of a screen coated with phosphor. The excited phosphors glow for a short period of time, then they must be recharged (refreshed). CRTs used in raster refresh or vector refresh devices are refreshed at least 30 times per second.

cursor.

A position indicator used to show where data will be entered on a display device. Usually represented by crosshairs or a block of blinking light.

diffused light source.

When rendering an object, the term used to describe a light source that is widespread. Multiple light sources or a light source that has been spread by the act of passing through a diffuser--the light source on a cloudy day.

digitizer.

The computer graphics hardware that is used to convert analog data into digital data--usually a graphics tablet. See digitizing.

digitizing.

The act of converting analog data into digital data. This is usually done with a graphics tablet. A drawing is taped to the tablet surface and the graphic image is traced with a puck or stylus. As the image is being traced, digital data is fed into the computer. 3D digitizing is also performed in some applications.

display buffer.

Also known as a frame buffer. That portion of the computer's RAM memory in a raster refresh computer graphics system that contains data needed to refresh the screen. A raster refresh display device's screen must be refreshed (redrawn) at least 30 times per second or the screen's image will fade and flicker will occur. Without a more permanent record of the screen's contents, an image could not be maintained on the screen long enough for the user to view it. See bit map.

display device.

That computer graphics hardware that produces the graphic image. The most common display devices are pen plotters, matrix printers, electrostatic plotters, and cathode ray tubes (CRTs); however, the term is used most frequently to refer to the CRT.

dot matrix printer.

An output device that is commonly used with personal computers. It is capable of creating text and graphics equally well. The image or text is created when the head (composed of a matrix of electronically controlled pins) strikes the ribbon and paper. Generally, the greater the number of pins in the matrix, the higher the quality of the printed image.

drum plotter.

A type of plotter (output device) that produces drawings on media that are fixed to a rotating cylinder. It can refer to any type of plotter that uses a rotating cylinder, but usually is used to refer to a pen plotter which uses a rotating cylinder.

electronic air brush.

That function in paint application software that simulates an air brush or spray can. An air brush sprays ink on the media for a unique effect in illustrating or painting.

electrostatic plotter.

A computer graphics output (display) device that generates an image by placing an electrostatic charge on the medium, in a raster format, and then passing the medium through a bath of toner particles. An operation very similar to that used by most plain paper copiers.

endpoint.

The extreme limits of a line, usually defined by Cartesian coordinates. The logical conclusion of an operation.

faceted polygons.

The plane surfaces that are used in computer graphics to render a solid figure. Rendering a figure in this manner results in an object that looks like a faceted gem.

fatal error.

A computer programming error that causes the program to quit executing.

file.

A collection of information (data) stored with a designated name on a mass storage device such as a disk.

flatbed pen plotter.

A computer graphics electromechanical output device where the media is in a stationary position on a flat surface and the image is drawn with a pen or pens held by a carriage traveling parallel to the surface.

flicker.

A condition that occurs on a CRT refresh display device when the refresh rate is slowed to the point where the human eye can detect the screen flashing. There is a noticeable decay in the luminance of the phosphors. This can happen on any refresh device; although, usually for different reasons.

flip book.

A book with an assembly of still images. By rapidly flipping through the book, an animated sequence can be viewed. A Kinetograph.

floppy diskette.

A magnetic mass storage device consisting of a flexible disk encased in a plastic or cardboard envelope. Used for random-access storage.

format a disk.

The task of having a disk prepared to receive data from a particular computing system. Each computer system formats (arranges) data on a disk in a unique manner. Therefore, a disk must be formatted on the computer system with which it is to be used.

FORTRAN.

A acronym that stands for FORmula TRANslation. A high level computer programming language that is very popular especially in computer graphics and engineering applications.

frame buffer.

A special portion of RAM memory designed to store a digital description of a computer image. Most usually used to describe the image being displayed on a raster refresh display device screen and from which the screen is refreshed.

function.

A process in an application program, such as drawing a line, that is executed when a specific command is given, a key is pressed, or a menu button is selected.

graphic transformation.

A modification made to a graphic image or its data base. The basic transformations are scaling, rotation, translation, reflection, perspective, and shearing.

graphics tablet.

A computer graphics input device consisting of a flat tablet and a stylus or puck. It converts graphical data (usually a drawing taped to the tablet surface) into binary data for the computer or it is used to control the screen cursor.

greeking.

The description given to illegible scrawl that results when an actual size desktop publishing page is changed to full page layout. The prose is reduced to a point that it is no longer legible.

grid.

A very common type of computer graphics input constraint. Input data is transformed (rounded off) to correspond to a grid intersection. The grid, any uniform pattern of dots or intersecting horizontal and vertical lines, can be displayed on the screen to aid the user. However, it is not necessary that the grid be displayed to have the input data transformed.

hard disk.

A fast computer mass storage device that can be housed in a separate case or within a computer. A single hard disk has a storage capacity of several million characters or bytes of information. This device utilizes a disk that is hard or firm as compared to a floppy disk.

hardware.

Physical computer equipment--electronic, magnetic, and mechanical components of a computer system. Contrasts with computer firmware, software, and courseware.

hexadecimal.

A numbering system with a radix of 16. The counting numbers in this system are 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E, and F. The hexadecimal number of 3B equals the decimal number of 59. $3 \times 16_{16} + B$ (or 11_{10}) $\times 1 = 59_{10}$.

high level language.

Any programming language such as BASIC, FORTRAN, PASCAL, and C that allows statements to be written in a familiar notation (English-like) code rather than machine or assembly code. High level language code must be translated to machine code before the computer can execute it. This is most usually done by a software interpreter or a compiler.

high resolution graphics display buffer.

See frame buffer.

I/O device.

I/O is an abbreviation for Input/Output. I/O devices are those hardware devices that are used with a computer system to allow data to be passed to or received from the central processing unit. Input devices used in computer graphics include light pens, tablets, mice, thumbwheels, track balls, touch screens, and joysticks. Graphic output devices include plotters, film recorders, matrix printers, laser printers, and various mass storage devices.

IGES.

An acronym that stands for Initial Graphics Exchange Specification. This is a standard adopted by many CAD vendors. The standard specifies how drawing data is to be formatted when the data is exchanged among the different CAD systems.

image enhancement.

The accentuation of a graphics image or part of an image by highlighting, zooming, blinking, or coloring.

input.

See I/O device.

interactive.

When used to describe a computer system, it is any system that yields an immediate response to the user's input. Where the computer and the user are in constant communication. Contrasts with batch processing system.

interpreter.

Software that acts as a language translator. It converts source language written in BASIC or another high level language into machine language code and it is executed immediately. Unlike a compiler, each statement is translated and executed before the next statement is processed.

isometric drawing.

A pictorial view of an object in which three faces of the object can be seen. This view is achieved by projection when the object is positioned so that its faces all make an angle of 35 degrees, 16 minutes to the projection plane. The resulting image's major axes are equally spaced (120 degrees apart) and equally foreshortened. However, when drawn by hand or a 2D computer graphics system, the axes are not usually foreshortened.

jaggies.

The stairstepping or saw-toothed effect achieved when drawing diagonal lines or simulated curved lines on a raster type device--a raster refresh display device, a matrix printer, or an electrostatic printer. The formal term describing this effect is aliasing.

joystick.

A computer graphics input device consisting of an upright stick and a button. Movement of the stick forward, backward, right, left, or in some combination manipulates the cursor on the screen. Very popular with computer or video games.

justification.

The act of arranging lines of digits or text to fit a prescribed pattern. Left justification results in groups of lines being flush on the left but with a jagged right edge. A justified group of lines has both right and left edges flush. Lines can also be center justified and right justified.

kerning.

A process used in printing where a piece of type with a certain character such as an L or a T is constructed in a manner to allow it to overlap the next piece of type (ligature). This results in printed copy with equal white space between characters, a result that cannot be achieved by just using proportional spacing.

kilobyte.

A measure of computing memory space. Specifically, 2 raised to the tenth power, or 1024 bytes. Kilo usually is thought to be 1000; however, in computing 1024 is the closest practical value. Therefore, 64K bytes is 64 x 1024 or 65,536 bytes of information. Abbreviated as kb. Compare to megabyte, gigabyte, and tenabyte.

layers.

A function within a CAD system that allows the user to segregate entities and then control them individually. They can be displayed or plotted together or separately. Layers can be thought of as separate sheets of tracing paper that are overlayed on a basic drawing. A technique used in architectural drawing when the basic floor plan is placed on one layer, the electrical plan on another, the plumbing plan on a third, etc.

leading.

Pronounced LED-ing. In printing and desktop publishing, refers to the spacing between lines of text. When all type was set by hand, spacing was achieved with thin strips of lead, thus the name.

letterspacing.

This is a technique used in printing where individual letters are slightly separated. It is the opposite of kerning.

light pen.

A graphics input device that is used in conjunction with the display screen. Shaped like a pen, it contains a photoelectric cell that detects light being emitted from the screen. With the proper software, it can be used to detect objects on the screen or draw images on the screen.

LOAD.

An Apple DOS and ProDOS command used to read a program from a file on a disk into the computer memory. Usually used in immediate mode. The key word LOAD is followed by the name of the filename under which the program is stored.

machine language.

The language understood by computers. Programs written in this language require no further interpretation in order for the machine to execute. Machine language instructions and data are presented in binary code.

mainframe.

A classification of computer central processing hardware representing computers that are very large and expensive. That classification between minicomputers and supercomputers.

mass storage device.

An output device that is capable of storing large amounts of data magnetically. It stores data in a non-volatile state. A term used to describe devices such as floppy disk drives, tape drives, and hard disk drives.

matrix printer.

A printer that is capable of printing text and graphics. It produces the image by using a printing head that is composed of a matrix of small pens capable of making dots when the head impacts the ribbon and paper. The pens can be selected in the proper arrangement to display any number, symbol, or alphabetic character.

matrix.

A two dimensional array of data. A matrix is a very convenient way to store data representing the coordinates of a graphics image.

megabyte.

A measure of computing memory space. Specifically, 2 raised to the 20th power or 1,048,576 bytes; 1024 kilobytes. See kilobyte elsewhere in this glossary.

menu.

A list of alternative processing functions from which the user can choose. In a computer graphics application, these functions can be listed as words or icons on the display device screen or they can be placed on a graphics tablet.

microcomputer.

A classification of computers. Usually the smallest and least expensive classification. It uses a microprocessor as a central processing unit.

microprocessor.

The central processing unit for a microcomputer. An integrated chip that contains arithmetic logic, control logic, and memory control elements necessary in a central processing unit.

minicomputer.

A classification of computer hardware. Refers to those computers that are smaller and less expensive than a mainframe, but larger and more expensive than workstations and microcomputers.

mirror.

Another term used to describe the basic transformation, reflection. A display of a graphic image in exactly the reverse orientation from its original position.

modeling.

The construction of a 3D object from 3D primitives (cylinders, cones, pyramids, prisms, etc.) by performing basic transformations such as scaling, translation, reflection, and rotation. Refer to wire frame, surface, and solid modeling.

mouse.

A graphic input device. Most commonly, a hand size housing containing a ball underneath and buttons on the top. The screen cursor is tracked by the movement of the device on a relatively small surface.

National Computer Graphics Association.

A professional association of computer graphics professionals from business, industry, and education. The association has headquarters in Arlington, Virginia and sponsors local chapters in many states.

object blend sequence.

The use of a computer to generate intermediate frames given key pictures or frames representing the extremities of a movement in an animation sequence.

object code.

The machine language computer code that is output from a compiler or assembler. It can be executed by the computer or is ready for further processing to produce executable code.

offset printing.

A method of printing that is very common. It is based on the principle that oil and water do not mix.

operating system.

That computer software that controls the basic operation of the computer. This software (sometimes referred to as OS) controls input and output (DOS), the execution of programs, debugging, data management, etc.

optical character readers.

A computer input system that accepts printed documents and identifies the characters by their shape. Some systems can only recognize characters when certain fonts have been used while others can even recognize handwriting.

origin.

The point where the X and Y axes of a Cartesian coordinate system intersect (X=0.0, Y=0.0, & if 3D Z=0.0). Usually the bottom left hand corner of the screen. However, Apple has chosen to use the upper left hand corner as the origin.

orthographic drawing.

A technical drawing that is derived by a form of projection where the projectors are parallel and make an angle of 90 degrees to the plane of projection. Types of drawings derived in this way are multiviews (2 and 3 views, etc.) and axonometrics (isometrics,dimetrics, and trimetrics).

output.

Data that is transferred to a storage device or some output device (plotter, printer, etc.) after being processed by the computer. Also, the final form of data that has been processed by a computer (a plot, a printout, an animated sequence).

Pascal.

A high level computer language named after the French mathematician, Blaise Pascal, who built the first desk-type-calculator adding machine in 1642. Pascal, the computing language, is an easy to use language that is highly structured and powerful. It is very popular in computer science college programs.

passive graphics.

A computer graphics program or application that transpires without intervention from the user. The opposite of interactive graphics.

pattern recognition.

The recognition of patterns, forms, or configurations by a computer system. The visual system used on a robot is an example of the practical use of such a system.

persistence of vision.

A physiological phenomenon which comes into play when humans watch T.V. or go to the movies. When pictures move faster than 18 to 24 times per second the images appear to blend together. This occurs because a single image flashed at the eye is retained longer by the brain than it is registered on the retina. The brain is still registering the last image when a new image is flashed; thus, the two images are blended together.

perspective drawing.

A type of technical or artistic drawing that most closely resembles how humans see images. This type of drawing is achieved by projecting an image on the plane of projection by projectors that are not parallel, but converge at some point designated as the station point (location of the observer). A type of drawing that is difficult to achieve on a drawing board, but relatively easy for a computer to execute.

photocopying.

The most popular method of document copying. Photocopiers can be found in most modern offices and are often referred to as plain paper copiers because special papers are not needed to achieve satisfactory copies. A copying process introduced by Xerox.

pixel.

The smallest element on a raster refresh type of computer display device that can be controlled by the computer. A raster refresh display device consists of a matrix (multiple rows and columns) of these elements. Individually, they can be turned off, turned on, or the color can be changed in order to draw different images. Short for "picture element."

point of intersection.

The position in space where a line crosses another line or the surface of a plane or solid.

polyline.

A graphic primitive or entity that consists of multiple lines joined end to end, but not closed so as to constitute a polygon.

program.

A series of instructions written for the computer which causes it to process data. The instructions can be written in many different types of languages from high level languages to machine languages.

PROM.

An acronym which stands for Programmable Read Only Memory. These chips are programmed after they are manufactured rather than at the time of manufacture as with ROM chips.

proofreading.

The task preformed in the publishing process just before the mass printing. Proofreaders examine the text carefully to find errors in spelling, grammar, and punctuation, as well as other details which make the final copy correct and easy to read.

puck.

A part of the graphics tablet input device. A graphics tablet can employ either a stylus or a puck for data input. The puck looks like a mouse and is moved over the tablet surface usually to control the cursor on the display device screen. The puck usually has a button or buttons on its top side and felt on its bottom.

RAM.

An acronym standing for Random Access Memory. This is the working memory of the computer; memory where user generated programs, application programs, and data are stored. RAM memory is volatile, that is, data and programs stored here are lost when the computer is turned off or when power is interrupted.

raster refresh.

The most common type of display device used in computer graphics. A cathode ray tube (CRT) device whose electronic beam excites the phosphors on the screen by scanning the screen in a fixed pattern--each row from left to right and from top to bottom. Picture elements (pixels) are lighted according to a corresponding memory location in the computer's frame buffer. This technology is very similar to commercial T.V.

real time animation.

Used to describe animation sequences that are presented by the computer at a speed that makes their movement appear as smooth and fluid as it would be in real life. The normal alternative would be to have individual frames transferred to motion picture film or another media and then presented at a rate greater than 18 to 24 frames per second. This term is sometimes used to refer to the animation that occurs with interactive applications such as flight simulation where new scenes must be generated according to the reaction of the user.

relative coordinates.

A set of Cartesian coordinates that are given in relation to the last set of coordinates addressed rather than in relation to the origin as with a set of absolute coordinates.

rendering.

An artist's representation of an image or the act of creating such a representation.

resolution.

The realism, fineness, and smoothness which can be achieved when an image is represented on a display device screen or another output device. With a raster type device, resolution is dependent on the density of or size of the pixels. The more dense, the smaller the pixels, or the more pixels or points per inch, the better the resolution.

ROM.

An acronym standing for Read Only Memory. A solid state chip on which data or programs are stored at the time of manufacture and cannot be changed by the user. These chips are non-volatile, that is, they do not loose the data stored on them when the power is turned off.

RUN.

An Apple command (also used in many other operating systems) that causes the program currently stored in RAM to be executed. If used with a line number following, the program begins executing with that line number. If used with the name of a program which is not presently in RAM, the program will be loaded from the diskette in the disk drive and then executed. An error will occur if DOS or ProDOS has not been loaded into the computer memory.

SAVE.

An Apple DOS or ProDos command that saves the program currently in RAM to the diskette in the designated disk drive. The name under which the program is saved must follow the command. The last disk drive used will be used unless the user specifies another. An error will occur if the user attempts to use this command when DOS or ProDOS has not been loaded into the computer's memory.

scanners.

A computer image processing input device. A scanner is used to transfer photos, sketches, maps, drawings, and text into digital data that can be used in drawing, painting, and desktop publishing application programs.

shaded images.

Two dimensional depictions of 3D objects where variations in the amount of light striking the object are emphasized to give the illusion of 3D.

simulation.

Simulation is the use of a computer to represent certain features of another physical or abstract system. One of the best known simulations performed by a computer is that of flight. Computer controlled flight simulators are used by the military and the major airlines for the training of pilots.

snap-to.

A type of input constraint used in graphics software such as CAD and desktop publishing applications. A location which is being input by the user is constrained to fall on a grid intersection or on a point that is part of an existing entity. An example of the latter would be when a location being input is constrained to fall on the center of an existing circle or the endpoint of an existing line. Applications usually allow the user to control the snap-to constraint.

software.

Sets of programs which are sets of instructions for the computer. The instructions that govern how the computer functions. Software can include application programs, system programs, and utilities.

solids modeling.

A method of designing objects that creates a data base which contains all the attributes necessary to describe the object as a 3D geometric solid. Other methods of modeling include wire frame and surface modeling.

source code.

Refers to a program in its original form before it has been processed by an assembler, compiler, or interpreter.

storage tube.

A type of computer graphics display device that was very popular before microcomputers were so prevalent. It used a CRT with very persistent phosphors or it saved the image as electrical charges --once an image was drawn it would remain on the screen indefinitely. Images were drawn using the directed beam method, that is, the beam followed the path of the vectors rather than traveling in a fixed path as with a raster refresh device. The storage tube had excellent resolution, was low cost, and compatible with time sharing systems.

storyboard.

A graphical outline of a film or computer generated animation sequence. A collection of single pictures, each of which represents a distinct sequence or narrative element within the animated sequence.

text fonts.

A term used in publishing to describe a complete set of alphabetic, numeric and special characters all of the same size and style (design). Examples: 12 pt. Helvetica, 14 pt. Times Roman, and 18 pt. Times Roman are all different fonts.

thaumatrope.

An early animation device invented by an English doctor, John Paris. Pronounced THAW-ma-trope, the device was a disk with strings attached with which to spin it and two pictures, one on each side. As the disk was spun between the strings, the two pictures appeared to be superimposed on each other.

thumbwheels.

A computer graphics input device used to control the screen cursor. Normally, there are two wheels, one to control the horizontal cursor movement and one to control the vertical cursor movement.

track ball.

A computer graphics input device used to control the screen cursor. Consists of a ball mounted in a housing. As the ball is rotated, the screen cursor moves at the speed and in the direction of the track ball. Technically, it is an upside down mouse.

transformation.

In computer graphics this term refers to changes made to the data base which describes the image being drawn. These changes are made mathematically. The results cause the image to be moved around on the screen, rotated, drawn larger or smaller. The basic transformations in computer graphics are scaling, rotation, translation, reflection, shearing, and perspective.

viewport.

A computer graphics term that refers to that portion of the display device screen that is being used to display the selected image (window).

window.

A computer graphics term used to describe that portion of the data base (image) that has been selected to be displayed within the display device's viewport.

wire frame model.

A computer graphics representation of a 3D image where lines are used to describe the edges of the solid figure or to form a wire mesh around a curved surface. The image appears in skeleton form, very often without hidden lines removed.

workstation.

A classification of computer between a microcomputer and a minicomputer. Usually refers to a 32 or 64 bit machine that is commonly used as an engineering workstation. Most workstations have enhanced graphics capabilities.

WYSIWYG.

An acronym that stands for What-You-See-Is-What-You-Get. A term used in desktop publishing to describe the capacity of the display device to closely display the same image that will be printed. Pronounced WHIZ-ee-wig.

X axis.

The horizontal base line in a Cartesian coordinate system. X coordinate distances are measured parallel and Y coordinate distances are measured perpendicular to this axis.

Y axis.

The vertical base line in a Cartesian coordinate system. Y coordinate distances are measured parallel and X coordinate distances are measured perpendicular to this axis.

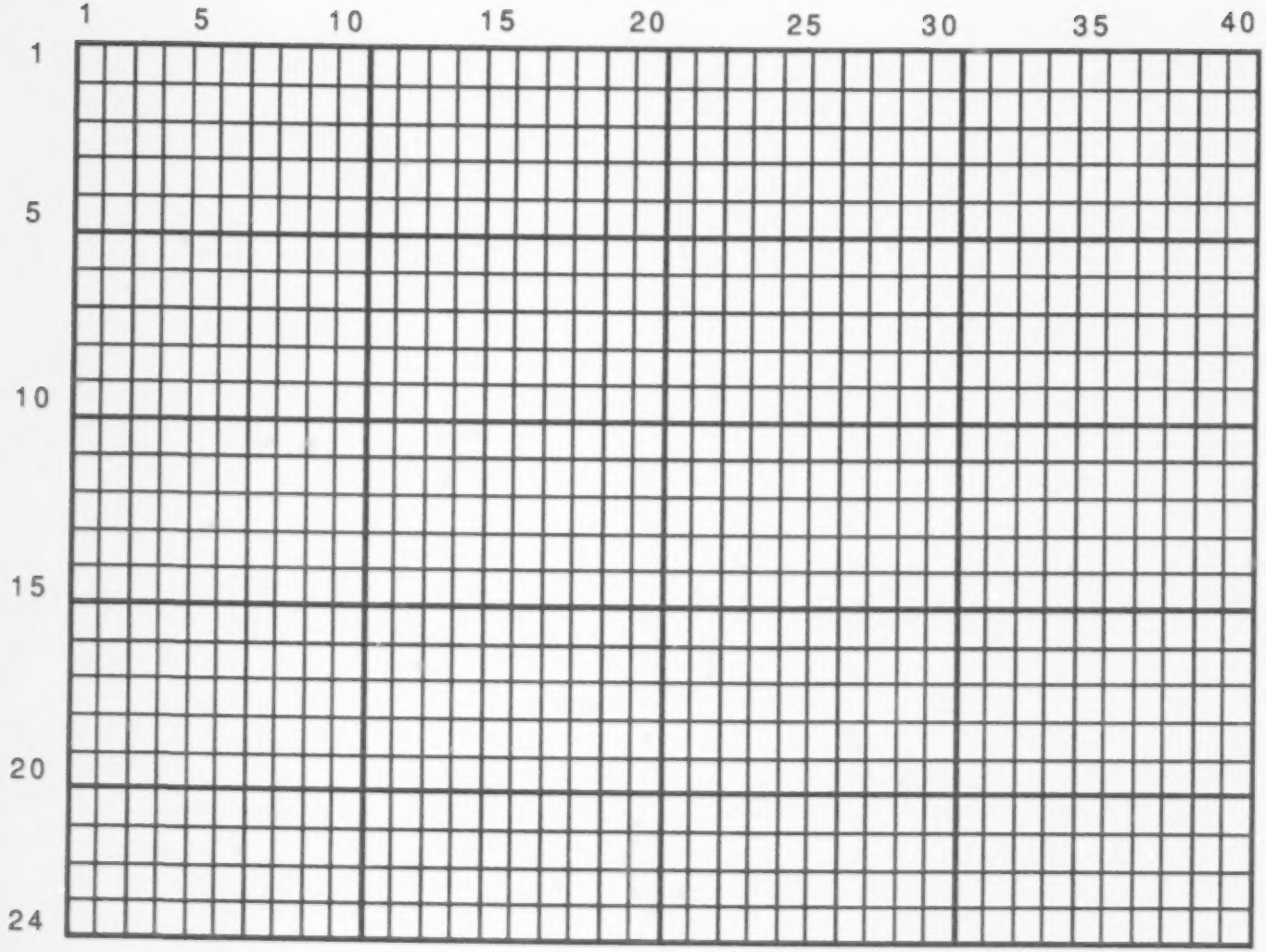
zoom.

The function in graphics application programs (CAD, Desktop Publishing, etc.) which moves the image closer or farther away from the user by increasing or decreasing the size of the window.

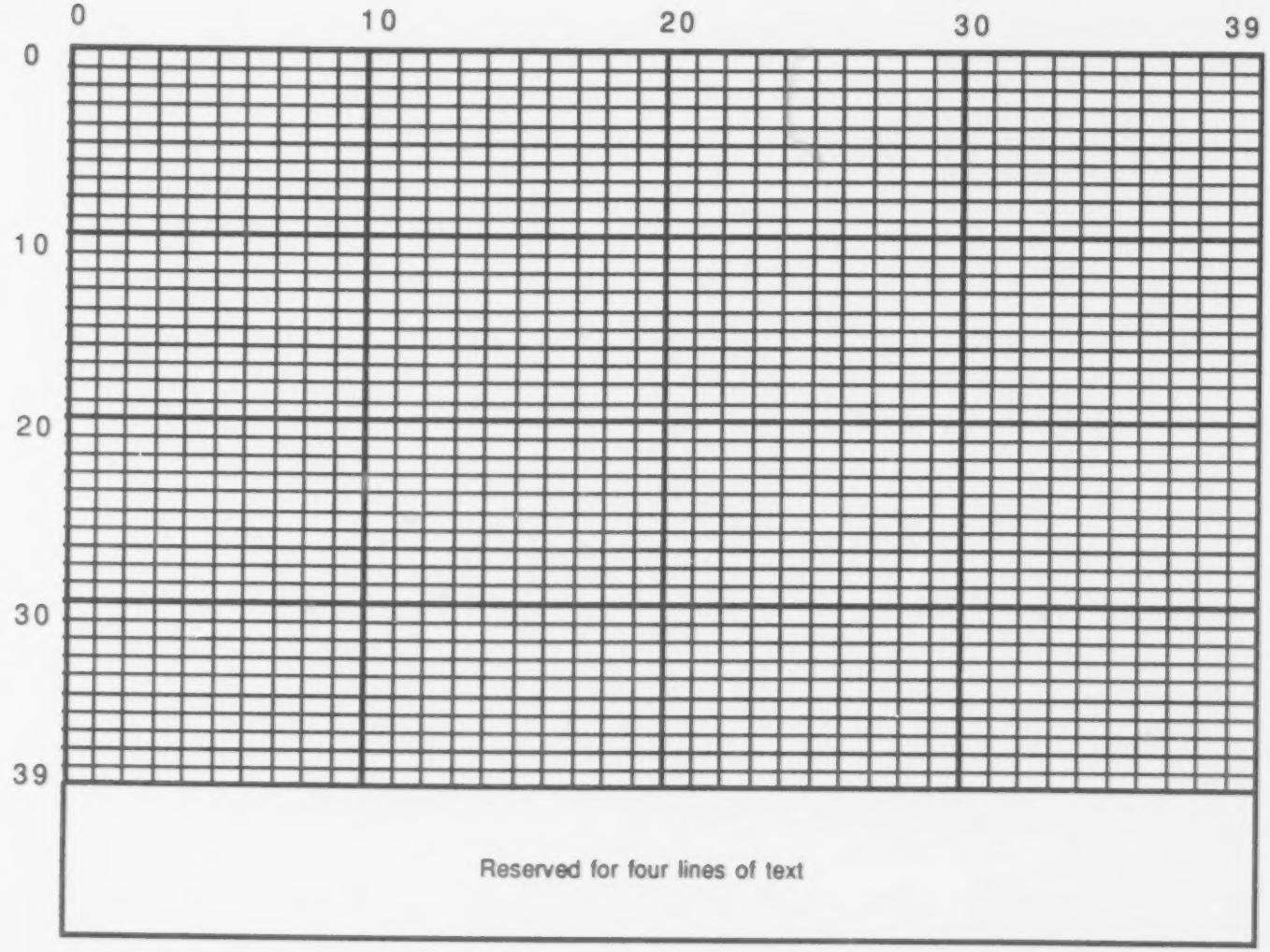
Appendix A

Apple Screen Grid Sheets

The Apple Text Screen

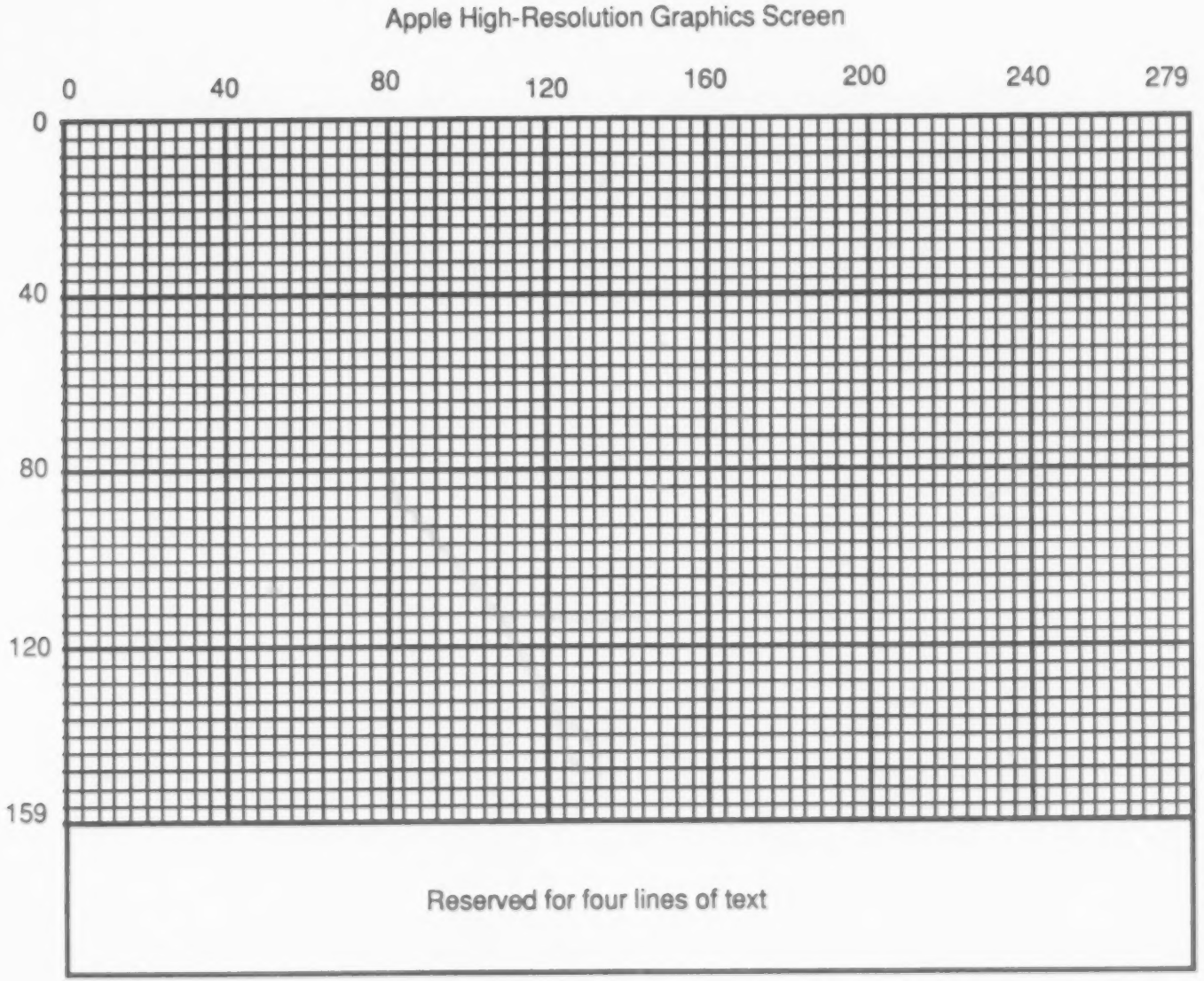


Apple Low-Resolution Graphics Screen



184

185


Apple High-Resolution Graphics Screen
0
40
80
120
160
200
240
279
0
40
80
120
159
Reserved for four lines of text

Appendix B

Equipment List

The following equipment is listed as an example of what could be purchased for a **new** laboratory. Individual schools will need to modify this list based on existing equipment and budgets. Before any purchases, teachers should check with their Apple dealer to learn about special educational discounts, new equipment that has been introduced since this list was compiled, and special hardware and software needed to connect and **network** this equipment.

Item	Number
Computer Workstation, Apple IIgs with 512k RAM, keyboard, mouse, Apple RGB monitor, 3.5 inch disk drive, and 5.25 inch disk drive	One per student + 1*
Printer, Apple Imagewriter II networked with 4 computers	One for every 4 computer workstations
Printer, Apple Imagewriter II	1*
Laser Printer, Apple Laserwriter II NT	1
Plotter, Hewlett Packard 7470A or 7475A	1
Tables, 3' x 4' computer	One per computer + 1 per printer + 2

* computer workstation and printer designated for the teacher.

If resources permit, consideration should be given to adding additional 3.5" disk drives and memory to the computers, modems, a scanner, and a large screen computer projection system to the above list.

Computer graphics laboratories can be set up using other Apple computers or other brands of computers. However, if this curriculum guide is to be used without modifications, Apple computers are the best selection and if new computers are being purchased the Apple IIgs is recommended.

Appendix C

Supply List

The following list of supplies is suggested for those laboratories where at least one of each type of equipment suggested in Appendix B is being used. Quantities ordered will depend on the size of the laboratory and number of classes being taught.

Item

Paper, Printer

Pen fed, 9.5" x 11" fan fold

8.5" x 11" copy paper, 16- to 24- pound bond

Paper, Plotter

"A" size (8.5" x 11"), 20 pound opaque white bond

"A" size, vellum

"A" size, polyester film

"B" size, 11" x 17", same as above (if lab is equipped with "B" size plotter).

Diskettes, Floppy

3.5" microdiskettes, double sided

5.25" single sided, double density

Pens, Plotter

Disposable fiber-tip for paper, black

Disposable fiber-tip for paper, assorted colors

Disposable fiber-tip for glossy film, black

Ribbons, Imagewriter II

Continous loop fabric ribbons, black

Continous loop fabric ribbons, color

Toner, Laserprinter

Appendix D

Facility Layout

A better but more complicated procedure is where a new end point is computed when one end point falls outside the window as in Figure 9. The procedure (algorithm) is changed to draw from the end point inside the window to a point on the edge of the window when only one end point is outside. This new point will be at the intersection of the original line and the edge of the window.

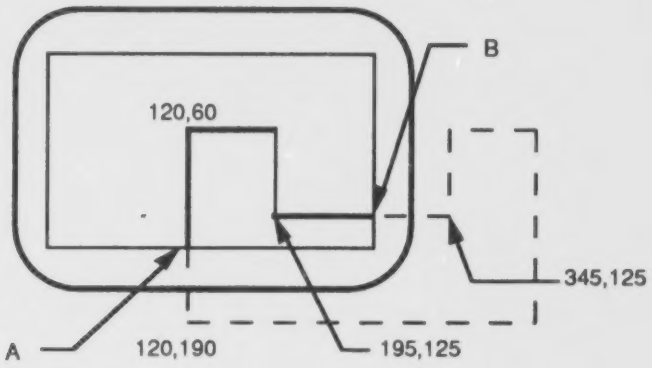


Figure 9.

Using this procedure (algorithm) on the image in Figure 9, new end points will be computed for the lines passing through points A and B. New coordinates of 120, 159 will be computed for point A, because the end point at 120, 190 has a Y coordinate larger than 159. New end point coordinates of 279, 125 are computed for point B, because the end point at 345, 125 has an X coordinate larger than 279.

Using this procedure to draw the rectangle, or any polygon with only horizontal or vertical lines, is not as complicated as it is for a polygon with inclined lines (see Figure 10). The logic that could be used in this special case (with only vertical or horizontal lines) is as follows:

a. Determine the coordinates of all the points in the polygon.
b. Examine the end points of each line in the polygon.
    1) If both end points are outside the window and the line does not pass through the window, do not draw the line and proceed to examine the next line.
    2) If both end points are outside the window and the line passes through the window, compute two new endpoints.
    2) Otherwise, examine each X and each Y coordinate as follows:
        (s) If an X coordinate is less than 0, then assign it a value of 0.
        (b) If an X coordinate is greater than 279, then assign it a value of 279.
        (c) If a Y coordinate is less than 0, then assign it a value of 0.
        (d) If a Y coordinate is greater than 159, then assign it a value of 159.
    3) Draw the line.

As an optional activity, change the program, [COG.008.4], in Part 4 to add a routine for clipping those lines that fall outside the window. The routine should work as described above.

The logic for a generalized procedure (algorithm) that could be used when drawing any polygon including the simulated circle and polygons with horizontal and vertical sides is too complicated for this course (see Figure 10). However, if you are an exceptional mathematics student and would like to learn more about it, you can read about clipping lines in Myers' *Microcomputer Graphics for the Apple Computer*.

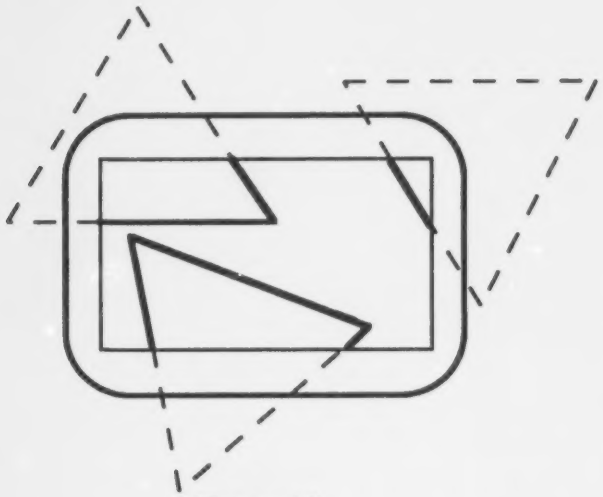


Figure 10.

Evaluation Criteria: To evaluate the student's success in completing this activity, the teacher should give the student a two part test. The first part should be a performance test where the student is required to write a high-resolution computer graphics program incorporating arrays, functions, and the BASIC commands DIM, GOSUB...RETURN and DEF FN. The student should be instructed to use only those commands covered in this activity or those covered in activities COG-004, COG-005, COG-006, and COG-007. The second part should be a recall test requiring the student to define the terms which were covered in this activity.

To be considered successful, the student's program should run on the computer without error and draw the intended image. On the second part, the student should be able to provide an acceptable definition for all of the terms. A term will be considered acceptable if it agrees with definitions found in current computer literature.

Computer Graphics Programming on an Apple II

GRAPHICS PROGRAMMING IN BASIC USING INPUT COMMANDS

Introduction: This activity is the sixth in a series of activities that is designed to teach the student how to draw computer images by programming an Apple computer with the BASIC language. This activity can be done on any Apple computer that uses Applesoft BASIC, the II+, IIe, IIc, or IIgs, and has an Apple mouse attached.

Enabling Objective: Before beginning this activity, the student should have completed Activity COG-008, Graphics Programming in BASIC Using Arrays and Subprograms. He or she should have demonstrated mastery of the objectives found in that activity before beginning this one.

Performance Objective (6-12 days):

TASK/COMPETENCY--COG-009

The student should be able to write an interactive graphics program which utilizes a graphics input device through the DOS and ProDOS commands IN# and PR#. He or she should also be able to define the following term relative to graphics programming:

input constraints

CONDITIONS

The above tasks should be performed under the following conditions:

a) The student should be provided an Apple computer with a disk drive, monitor, mouse (preferably an Apple mouse) and a printer and an Applesoft BASIC reference manual when writing the graphics program.

b) The student should be expected to recall the definition of the above term without the aid of a reference.

STANDARD OF PERFORMANCE

The above tasks should be performed to the following standards:

a) The graphics program should incorporate high-resolution graphics commands and those other Applesoft commands listed above or covered in COG-004, COG-005, COG-006, COG-007, and COG-008 activities. It should run without error and permit the user to input graphical data by using an Apple mouse or other input device.

b) The student should be able to supply a correct definition for the term that is similar to a definition found in the current computer literature.

Selected Instructional Materials: An Apple II+, IIe, IIc, or IIgs computer with printer, monitor (preferably with color), input device (preferably an Apple mouse), and disk drive; an Applesoft reference manual; an Apple II utilities diskette or system master diskette; a formatted diskette; the Computer Graphics Curriculum Project Master Diskette; a piece of graph paper representing the Apple high-resolution graphics screen; and a pencil.

Teacher References

Recommended

Books:

Apple Computer, Inc. (1983). *Applesoft BASIC programmer's reference manual.* Reading, MA: Addison-Wesley.

Apple Computer, Inc. (1983). *Apple II DOS user's manual.* Cupertino, CA: Author.

Apple Computer, Inc. (1987). *BASIC programming with ProDOS.* Reading, MA: Addison-Wesley.

Software:

Apple IIgs System Disk, ProDOS User's Disk, or *Apple II System Master (DOS 3.3).*

Teacher Activities

1. Familiarize yourself with the Applesoft BASIC library function CHR$ and the DOS and ProDOS commands IN# and PR#.

2. Look up the meaning of the term input constraint; review the student activities to see how it fits into the activities; and then explain it to the students. Be sure to explain the different types--angular, linear, grid, etc.

3. Demonstrate to the students how to use DOS or ProDOS commands in an Applesoft BASIC program. The following are some reminders:

   a. Before explaining the function CHR$, be sure the students understand ASCII codes. Show them an ASCII code table and explain how these codes (decimal) can be used with the CHR$ function.

   b. Even if you have covered it before, show them the slots on the Apple motherboard and how these relate to the IN# and PR# commands.

Student References

Recommended

Books:

Apple Computer, Inc. (1983). *Applesoft BASIC programmer's reference manual.*Reading, MA: Addison-Wesley.

Apple Computer, Inc. (1983). *Apple II DOS user's manual.* Cupertino, CA: Author.

Apple Computer, Inc. (1987). *BASIC programming with ProDOS.* Reading, MA: Addison-Wesley.

Software:

Apple IIgs System Disk, ProDOS User's Disk, or *Apple II System Master (DOS 3.3).*

Student Activities:

1. Read about the Applesoft BASIC function CHR$ and the DOS and ProDOS commands IN# and PR# in one of the above references. Then, load the HELLO program you wrote to INITialize your program diskette in Activity COG-003.

   old program from Activity COG-003:

   [COG.003.1]

```
10 REM  ** HELLO PROGRAM **
20 PRINT "COMPUTER GRAPHICS"
30 PRINT "PROGRAMS"
40 PRINT: PRINT
50 PRINT "student's name"
60 PRINT "school's name"
70 PRINT "date"
80 END
```

   Add the following lines to cause the program to give the user a list of the programs on the disk each time it is booted. As you can see, line 150 uses the CHR$ function and the DOS/ProDOS command CATALOG to provide the listing.

   [COG.009.1]

```
80 REM ***********************
90 REM  * GIVES  THE  USER    *
100 REM * A  LIST  OF  THE    *
110 REM * DISK'S  PROGRAMS *
120 REM ***********************
130 PRINT:PRINT
140 PRINT "PRESS ANY KEY TO CONTINUE THE LISTING"
150 PRINT CHR$ (4); "CATALOG"
160 END
```

If you saved this program originally with the name HELLO, use that name now to save it. Originally, you saved it as you formatted your disk with the INIT command. **Do not use the INIT command now or you will erase your disk. Use the SAVE command.**

You can use this same programming procedure and incorporate any of the DOS/ProDOS commands (RUN, LOAD, LIST, etc.) into your BASIC programs.

2. Review the INPUT command from Activity COG-005 and then key the following program into your computer. This program turns the mouse on, calls for input from the mouse, prints the values obtained from the mouse on the screen, and will stop when you press and release the mouse button.

[COG.009.2]

```
10 REM **  ACTIVITY  NINE    **
20 REM * COMPUTER GRAPHICS *
30 REM ************************
40 REM * ALLOWS FOR  VALUES *
50 REM * TO  BE  INPUT  FROM   *
60 REM * THE  APPLE  MOUSE    *
70 REM ************************
80 GOSUB 1000
90 INPUT "";X,Y,A
100 PRINT X,Y,A
110 IF A = 3 GOTO 130
120 GOTO 90
130 GOSUB 2000
140 TEXT:HOME
150 END


1000 REM ** SUBROUTINE   TO **
1010 REM ** TURN  MOUSE  ON **
1020 REM ***********************
1030 LET D$ = CHR$ (4)
1040 PRINT D$;"PR#4": PRINT CHR$ (1)
1050 PRINT D$;"PR#0"
1060 PRINT D$;"IN#4"
1070 RETURN
1080 END

2000 REM ** SUBROUTINE   TO **
2010 REM ** TURN  MOUSE  OFF *
2020 REM ***********************
2030 PRINT D$;"IN#0"
2040 PRINT D$;"PR#4":PRINT CHR$ (0)
2050 PRINT D$;"PR#0"
2060 RETURN
2070 END
```

RUN, debug, and SAVE this program with a unique name (COG.009.2).

When you run this program, it should print an endless stream of lines each with three values (see lines 90-120). The first two values will range from 0 to +1023 and the third value will be either a 1,2,3, or 4. The first of the values represents the X position of the mouse; the second, the Y position of the mouse; and the third, the status of the button on the mouse. If the third value is a 1, it means that the mouse button is down and was down during the last INPUT statement. If the third value is a 2, it means that the mouse button is down and was up during the last INPUT statement. If the third value is a 3, it means that the mouse button is up and was down during the last INPUT statement. A value of 4 means that the button is up and was up during the last INPUT statement. Try to change the values by moving the mouse around and by pressing the mouse button. If you press it and then release it in the right way you will obtain a value of 3 and the program will stop (see the IF statement at line 110).

The subroutine at lines 1000-1080 turns the mouse on. In line 1030, a CTRL D character (ASCII character #4) is assigned to the variable D$. This is to save space, since a CTRL D must precede any DOS or ProDOS command in a PRINT statement. It also makes the program readable. A CTRL D could be keyed into every print command without using the CHR$ function; however, there is nothing printed on the screen when a CRTL D is keyed. Therefore, it is invisible to someone reading your program.

At line 1040, the DOS/ProDOS command PR#4 is used to send subsequent output through slot 4, the mouse port. This is followed on the same line with a CTRL A, the command that turns the mouse on. The DOS/ProDOS command at line 1050, PR#0, directs output back to the monitor screen. The command, IN#4, at line 1060 directs future INPUT commands to take data from slot #4, the mouse port.

The subroutine that turns the mouse off (lines 2000-2070) reverses those things accomplished in the previous subroutine. The DOS/ProDOS command, IN#0, at line 1030 directs the computer to take future INPUT data from the keyboard. The two statements at line 1040 send a CRTL @ to the mouse through slot 4. This turns the mouse off. And, the line at 1050 directs output back to the monitor screen (undoing the effect of the PR#4 in the previous line).

3. Use the subroutines from the previous program, but change the main program as follows to make it a graphics program that allows the user to paint with the mouse:

[COG.009.3]

```
10 REM **   ACTIVITY  NINE   **
20 REM * COMPUTER GRAPHICS *
30 REM ************************
40 REM * ALLOWS FOR  INPUT  *
50 REM * FROM  THE APPLE    *
60 REM * MOUSE FOR PAINTING*
70 REM ************************
75 REM * TURN MOUSE ON,     *
80 REM * PUT IN GRAPHICS    *
85 REM * MODE, & OBTAIN     *
90 REM * INPUT FROM MOUSE  *
95 REM ************************
100 GOSUB 1000
110 HGR:HCOLOR=3
120 INPUT "";X,Y,A
130 REM ************************
140 REM * CONSTRAIN  INPUT   *
```

```
150 REM * TO  FALL  WITHIN     *
160 REM * APPLE  WINDOW       *
170 REM ************************
180 IF X > 279 THEN X = 279
190 IF Y > 159 THEN Y = 159
200 REM ************************
210 REM * IF  MOUSE  BUTTON    *
220 REM * IS  DOWN  PLOT  A    *
230 REM * POINT;  DO  UNTIL    *
240 REM * MOUSE  BUTTON HAS  *
250 REM * BEEN PUSHED  TWICE*
260 REM ************************
270 IF A = 1 THEN HPLOT X,Y
280 IF A = 2 AND B = 3 AND C = 2 GOTO 320
290 LET C = B:LET B = A
300 FOR W = 1 TO 100:NEXT W
310 GOTO 120
320 GOSUB 2000
330 TEXT:HOME
340 END


1000 REM ** SUBROUTINE   TO **
1010 REM ** TURN  MOUSE  ON **
1020 REM ************************
1030 LET D$ = CHR$ (4)
1040 PRINT D$;"PR#4": PRINT CHR$ (1)
1050 PRINT D$;"PR#0"
1060 PRINT D$;"IN#4"
1070 RETURN
1080 END

2000 REM ** SUBROUTINE   TO **
2010 REM ** TURN  MOUSE  OFF *
2020 REM ************************
2030 PRINT D$;"IN#0"
2040 PRINT D$;"PR#4":PRINT CHR$ (0)
2050 PRINT D$;"PR#0"
2060 RETURN
2070 END
```

Before you RUN this program, you should understand the logic of it and what is expected from the user. Lines 110-120 in the main program turn the mouse on (by calling the subroutine at 1000), put the computer in high-resolution graphics mode, set the color to white, and ask for INPUT. The subroutine at 1000 has set the computer to accept INPUT through slot #4; therefore, INPUT will come from the mouse rather than the keyboard. The quote marks in the INPUT statement are there to suppress the question mark that would appear on the screen without them.

Remember, in the previous program the values returned from the mouse ranged from 0 to +1023. If the mouse values are to be used to HPLOT on the Apple high-resolution graphics screen, they must be changed so that they do not exceed 279 for X values and 159 for Y values. This is what is being done at lines 180 and 190. This is an example of a *input constraint*. Get your instructor to explain *input constraints* in greater detail.

At line 270, the program HPLOT's a point if the mouse button is being held in the down position. Remember, if the third value (the variable A) is a 1, it means that the mouse button is down and was down during the last INPUT statement.

The function of the statements at lines 280 and 290 is to examine the status of the button during the last three INPUT's. The statements at line 290 store the values of variable A during the last two INPUT's. These values are stored in variables B and C. The IF statement at line 280 examines them along with the current value of A. When the three values indicate that the button has been pushed twice, the program will be sent to line 320 and the program will come to an END. Please note that to have the three values 2,3,and 2 returned in that order, the mouse button must be pushed (double-clicked) very quickly. If it doesn't work for you after several attempts, change the FOR..TO and NEXT loop at line 300. This is a pause loop to allow the computer to read your double click.

RUN, debug, and SAVE this program with a unique name (COG.009.3).

4. Try changing the shape of the paint brush in the previous program by changing line 270 as follows:

[COG.009.4]

from:

```
270 IF A = 1 THEN HPLOT X,Y
```

to:

```
270 IF A = 1 THEN HPLOT X,Y+4 TO X+4,Y
```

To prevent an error message because of an unacceptable X or Y value with the HPLOT command, you should also change lines 180 and 190.

from:

```
180 IF X > 279 THEN X = 279
190 IF Y > 159 THEN Y = 159
```

to:

```
180 IF X > 275 THEN X = 275
190 IF Y > 155 THEN Y = 155
```

Try other paint brushes of different sizes and shapes.

SAVE these versions of the program with unique names (COG.009.4).

5. For most graphic applications it is desirable to have a cursor appear on the screen that tracks the relative position of the mouse. Key the following main program into your computer. It contains such a routine. The routine contains a FOR..TO and NEXT loop that continually draws and then erases two lines that intersect at the current cursor position.

[COG.009.5]

```
10 REM **  ACTIVITY  NINE   **
20 REM * COMPUTER GRAPHICS *
30 REM ************************
40 REM * ALLOWS FOR INPUT  *
50 REM * FROM THE APPLE    *
60 REM * MOUSE FOR PAINTING*
70 REM ************************
80 PRINT "TO PAINT, USE MOUSE."
90 PRINT "TO QUIT, PRESS ANY"
100 PRINT "KEY ON KEYBOARD."
110 FOR W = 1 TO 9000:NEXT W
120 REM ************************
130 REM * TURN MOUSE ON,   *
140 REM * PUT IN GRAPHICS  *
150 REM * MODE, & OBTAIN   *
160 REM * INPUT FROM MOUSE *
170 REM ************************
180 GOSUB 1000
190 HGR:HCOLOR = 3
200 INPUT "";X,Y,A
210 REM ************************
220 REM * CONSTRAIN  INPUT  *
230 REM * TO  FALL  WITHIN   *
240 REM * APPLE  WINDOW      *
250 REM ************************
260 IF X < 6 THEN X = 6
270 IF X > 273 THEN X = 273
280 IF Y < 5 THEN Y = 5
290 IF Y > 154 THEN Y = 154
300 REM ************************
310 REM * DRAWS THE CURSOR *
320 REM ************************
330 FOR M = 1 TO 2
340 HPLOT X,Y-5 TO X,Y+5
350 HPLOT X-6,Y TO X+6,Y
360 HCOLOR = 0
370 NEXT M
380 HCOLOR = 3
390 REM ************************
400 REM * IF MOUSE BUTTON   *
410 REM * IS DOWN PLOT A    *
420 REM * POINT; DO UNTIL   *
430 REM * ANY KEY ON KEY-   *
440 REM * BOARD IS PRESSED  *
450 REM ************************
460 IF A = 1 THEN HPLOT X,Y
```

```
470 IF A < 0 GOTO 490
480 GOTO 120
490 GOSUB 2000
500 TEXT:HOME
510 END
```

The differences between this program and the one in Part 3 (COG.009.3) are these:

The IF..THEN statements at lines 180 and 190 were changed and two more IF ..THEN statements were added at lines 200 and 210. These changes were made to allow for the cursor dimensions (see the following change).

A routine was added (lines 300-380) to draw a cursor. The cursor is a horizontal line 12 pixels long intersecting at the midpoint with a vertical line 10 pixels long. The cursor is continually drawn (in white) and erased (redrawn in black) to coincide with each new position that is being input. If the cursor was not erased, there would be a path of cursors left across the screen similar to the paintbrush in the previous program. An HCOLOR = 3 statement was added at line 300 because when the program falls out of the loop, HCOLOR = 0 (black).

Notice that this program uses another method for the user to exit. When receiving INPUT from the mouse, a negative number in position three (our variable A) indicates that a key on the keyboard has been hit. As you can tell, this feature was used in this program to allow the user to exit by way of the keyboard (see line 470). Directions to indicate this difference were added with PRINT statements at lines 80-100. A pause loop, to allow the user time to read the directions, was added at line 110.

The statement at line 460 was changed to draw only one point when the mouse button is pushed.

The program was renumbered.

Of course, the subroutines to turn the mouse on and off are still needed. They should not be changed.

```
1000 REM ** SUBROUTINE   TO **
1010 REM ** TURN  MOUSE  ON **
1020 REM ************************
1030 LET D$ = CHR$ (4)
1040 PRINT D$;"PR#4": PRINT CHR$ (1)
1050 PRINT D$;"PR#0"
1060 PRINT D$;"IN#4"
1070 RETURN
1080 END

2000 REM ** SUBROUTINE   TO **
2010 REM ** TURN  MOUSE  OFF *
2020 REM ************************
2030 PRINT D$;"IN#0"
2040 PRINT D$;"PR#4":PRINT CHR$ (0)
2050 PRINT D$;"PR#0"
2060 RETURN
2070 END
```

RUN, debug, and SAVE this program with a unique name (COG.009.5).

Next, make the following changes in the program. These will allow the user to draw horizontal lines.

[COG.009.5.1]

change:

from:

```
80 PRINT "TO PAINT, USE MOUSE."
```

to:

```
80 PRINT "TO DRAW, USE MOUSE."
```

add:

```
185 DIM XX(2), YY(2)
195 LET N = 1
```

DELete lines 410-510 and replace with:

```
410 REM * IS  PUSHED,  RECORD *
420 REM * POINT.  AFTER  TWO  *
430 REM * PTS  ARE  INPUT,     *
440 REM * DRAW  A  LINE.  STOP*
450 REM * WHEN ANY  KEY ON  *
460 REM * KEYBOARD  IS  USED *
470 REM **********************
480 IF A = 3 THEN XX(N) = X:LET YY(N) = Y:LET N = N + 1
490 IF N = 3 THEN HPLOT XX(1),YY(1) TO XX(2),YY(1):LET N = 1
500 IF A < 0 GOTO 520
510 GOTO 120
520 GOSUB 2000
530 TEXT:HOME
540 END
```

The above program illustrates another example of an input constraint. In this program, all lines are constrained to be horizontal. Without constraints, it would be difficult to draw with a computer. You might think of constraints as the T squares and straight edges of computer graphics.

In line 480 above, several things happen if the button has been pushed on the mouse. The values of X and Y (the current cursor position) are stored in XX(1), YY(1) or XX(2), YY(2) depending on whether this is the first point input or the second. Also, the variable N is increased by increments of one. Even though there are three statements on this line and only the first is an IF..THEN, none of the statements will be executed if the IF..THEN is false. This is a unique characteristic of Applesoft BASIC. This will probably not work in the same way with other BASIC languages.

The statements at line 490 draw a line and set N back to 1, if two points have been input. As with line 480, the subsequent statement (LET N = 1) is not executed if the IF...THEN is false. Notice how YY (1), the Y value of the first point is also used as the Y value for the second endpoint. This is what keeps the line horizontal.

By making the following change the program will draw vertical lines instead of horizontal lines:

[COG.009.5.2]

from:

```
490 IF N = 3 THEN HPLOT XX(1),YY(1) TO XX(2),YY(1):LET N = 1
```

to:

```
490 IF N = 3 THEN HPLOT XX(1),YY(1) TO XX(1),YY(2):LET  N = 1
```

6. The cursor in the above program has at least two faults. It flickers too much and it acts as an eraser when you cross over an object that has been drawn. You can improve on the first fault by drawing the cursor using an Applesoft shape table and the commands, DRAW, XDRAW, and SIZE. You can do this as an optional activity. The second fault is best solved by using a machine language routine with an exclusive OR operation to draw the cursor. The latter is not a recommended activity for this course.

7. Change the program used in part 5, above so that it draws a rectangle. The program should have the user designate the diagonal corners of the box by two inputs with the mouse.

8. As an optional activity, design a program that draws circles based on points input with the mouse.

Evaluation Criteria: To evaluate the student's success in completing this activity, the teacher should give the student a two part test. The first part should be a performance test where the student is required to write a high-resolution computer graphics program incorporating DOS and ProDOS commands, especially IN# and PR#. The program should incorporate an Apple mouse or other input device. The student should be instructed to use only those commands covered in this activity or those covered in activities COG-004, COG-005, COG-006, COG-007, and COG-008. The second part should be a recall test requiring the student to define the term, input constraint.

To be considered successful, the student's program should run on the computer without error, allow input from the mouse or other device, and draw the intended image. On the second part, the student should be able to provide an acceptable definition for the term. It will be considered acceptable if it agrees with a definition found in current computer literature.

Computer Graphics Programming on an Apple II

GRAPHICS PROGRAMMING IN BASIC USING OUTPUT COMMANDS

Introduction: This activity is the seventh in a series of activities that is designed to teach the student how to draw computer images by programming an Apple computer with the BASIC language. This activity can be done on any Apple computer that uses Applesoft BASIC, the II+, IIe, IIc, or IIgs, and has a pen plotter attached (preferably a Hewlett Packard plotter or one that uses the Hewlett Packard plotting language, HP GL).

Enabling Objective: Before beginning this activity, the student should have completed Activity COG-009, Graphics Programming in BASIC Using Input Commands. He or she should have demonstrated mastery of the objectives found in that activity before beginning this one.

Performance Objective (6-12 days):

TASK/COMPETENCY--COG-010

The student should be able to write a graphics program which utilizes a graphics output device through the DOS or ProDOS commands IN# and PR# and the HP GP plotter language. The student should also be able to define the following terms relative to graphics programming:

viewport
absolute coordinates
relative coordinates
"A", "B", and "C" size paper

CONDITIONS

The above tasks should be performed under the following conditions:

a) The student should be provided an Apple computer with a disk drive, monitor, a Hewlett Packard pen plotter, an Applesoft BASIC reference manual, and the Hewlett Packard *Interfacing and Programming Manual* (for the appropriate HP plotter) when writing the graphics program.

b) The student should be expected to recall the definition of the above terms without the aid of a reference.

STANDARD OF PERFORMANCE

The above tasks should be performed to the following standards:

a) The graphics program should incorporate high-resolution graphics commands and those other Applesoft commands listed above or covered in COG-004, COG-005 COG-006, COG-007 , COG-008, or COG-009 activities. It should run without error and permit the user to input graphical data by using an Apple mouse or other input device.

b) The student should be able to supply a correct definition for all of the terms. Definitions should be considered correct if they conform to definitions found in the current computer literature.

Selected Instructional Materials: An Apple II+, IIe, IIc, or IIgs computer with printer, monitor (preferably with color), pen plotter (preferably a Hewlett Packard pen plotter), and disk drive; an Applesoft reference manual; the Hewlett Packard *Interfacing and Programming Manual* for the plotter that is to be used; and an Apple II utilities diskette or system master diskette; and a formatted diskette.

Teacher References

Recommended

Books:

Apple Computer, Inc. (1983). *Applesoft BASIC programmer's reference manual.* Reading, MA: Addison-Wesley.

Apple Computer, Inc. (1983). *Apple II DOS user's manual.* Cupertino, CA: Author.

Apple Computer, Inc. (1987). *BASIC programming with ProDOS.* Reading, MA: Addison-Wesley.

Hewlett Packard. (1983). *Interfacing and programming manual.* San Diego, CA: Hewlett Packard.

Software:

Apple IIgs System Disk, ProDOS User's Disk, or *Apple II System Master (DOS 3.3).*

Teacher Activities

1. The Applesoft BASIC library function CHR$ and the DOS and ProDOS commands IN# and PR# that were introduced in COG-009 will also be used extensively in this activity. Review these commands again, if you need to.

2. The terms viewport and window which were introduced earlier, should be reviewed again with the student.

3. Familiarize yourself with the Hewlett Packard Graphics Language (HP GL) which is explained in the *Interfacing and Programming Manual*. Read Chapters 1-5 carefully. The following HP GL mnemonic commands are important to this activity: AA, AR, CI, CP, DT, IN, IP, IW, LB, LT, PA, PD, PR, PU, SC, SI, SP, and SR. Make special note of them when reading these chapters.

4. Be certain that the plotter has been properly installed and is in working condition. The *Interfacing and Programming Manual* and the operating manual that came with your plotter should be consulted when connecting your plotter. Run a test program on the plotter before the students are ready to use it.

5. Be prepared to offer suggestions for the selection of an object to be drawn in the Student Activity Part 3. The object should require the student to plot circles and arcs.

Student References

Recommended

Books:

Apple Computer, Inc. (1983). *Applesoft BASIC programmer's reference manual*. Reading, MA: Addison-Wesley.

Apple Computer, Inc. (1983). *Apple II DOS user's manual*. Cupertino, CA: Author.

Apple Computer, Inc. (1987). *BASIC programming with ProDOS*. Reading, MA: Addison-Wesley.

Hewlett Packard. (1983). *Interfacing and programming manual*. San Diego, CA: Hewlett Packard.

Software:

Apple IIgs System Disk, ProDOS User's Disk, or Apple II System Master (DOS 3.3).

Student Activities:

1. Review again those commands introduced in COG-009--the Applesoft BASIC function CHR$ and the DOS or ProDOS commands IN# and PR#. Then, read Chapters 1-5 in the *Interfacing and Programming Manual*. Make special note of the HP GL commands IN, IP, PA, PD, PU, SC, and SP. When you feel comfortable with these commands, try them in the following program. It draws a border line on a piece of "A size" (8.5 x 11 inch) paper. Therefore, before you RUN this program, be sure you are working at a computer that has a Hewlett-Packard plotter properly attached. You will also need to turn the plotter on, place pens in the pen holders, and have a sheet of 8.5" x 11" ("A size") paper properly loaded.

COG.010.1

```
10 REM **    ACTIVITY TEN    **
20 REM * COMPUTER GRAPHICS *
30 REM ************************
40 REM * PLOTS A BORDER      *
50 REM * USING THE HEWLETT *
60 REM * PACKARD PLOTTER    *
70 REM ************************
80 REM * SELECT SLOT FOR     *
90 REM * PLOTTER OUTPUT      *
95 REM ************************
100 LET D$ = CHR$ (4)
110 PRINT D$; "PR#2"
120 REM ************************
130 REM * SEND PLOTTER CMDS*
140 REM * TO INITIATE, AND    *
150 REM * TO SET VIEWPORT    *
160 REM * AND WINDOW          *
170 REM ************************
180 PRINT "IN;"
190 PRINT "IP 0,0,10100,7140;"
200 PRINT "SC 0,10,0,7.5;"
210 REM ************************
220 REM * SEND PLOTTER CMDS*
230 REM * TO SELECT PEN,      *
240 REM * PLOT ABSOLUTE, &   *
250 REM * PLOT BORDER W/PEN*
260 REM * UP AND PEN DOWN *
270 REM ************************
280 PRINT "SP 1;"
290 PRINT "PA,PU 0,0 PD 9.8,0 9.8,7 0,7 0,0;"
600 REM ************************
610 REM * RETURN PEN TO      *
620 REM * HOLDER & CEASE     *
630 REM * OUTPUT TO SLOT # 2 *
640 REM ************************
650 PRINT "SP 0;"
660 PRINT D$;"PR#0"
670 END
```

RUN, debug, and SAVE this program with a unique name (COG.010.1).

In the above program, output to the plotter is directed through slot #2. If your plotter is connected to a card in a different slot, you will need to change the PR# command in line 110 accordingly. Plotter commands begin at line 180 with the command to initiate the plotter and cease at line 650 with the command to select pen 0 (return the pen to its holder). Notice that these commands and all intermediate plotter commands end with a semi-colon. If they did not, a carriage return would be output and the plotter commands would be terminated. The IP command at line 190 is used to select the viewport, and the SC command at line 200 is used to select the window. The same considerations that have been used in previous activities when selecting a viewport and a window need to be used here, that is, the aspect ratios of the viewport and the window need to be the same to avoid distortion of the image being drawn.

2. Read about the HP GL commands DT, LB, LT, PR, and SR. Using the above program, COG.010.1, as a skeleton program, add the necessary commands using lines beginning with 300 but smaller than 600 to draw the image shown below.

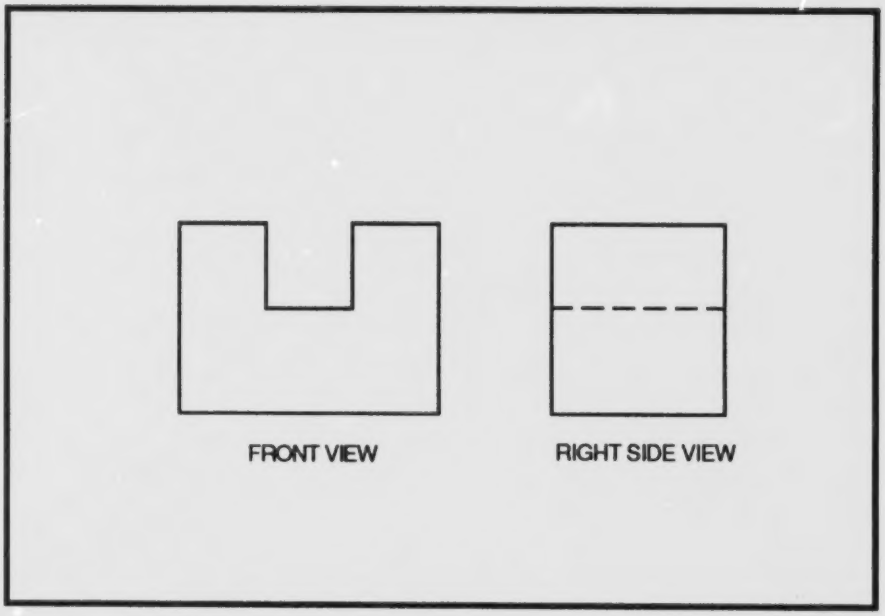


Figure 11.--Two View Drawing

[COG.010.2]

```
10 REM **   ACTIVITY TEN    **
20 REM * COMPUTER GRAPHICS *
30 REM ************************
40 REM * PLOTS TWO VIEWS    *
50 REM * USING THE HEWLETT *
60 REM * PACKARD PLOTTER    *
70 REM ************************
80 REM * SELECT SLOT FOR    *
90 REM * PLOTTER OUTPUT     *
95 REM ************************
100 LET D$ = CHR$ (4)
110 PRINT D$; "PR#2"
120 REM ************************
130 REM * SEND PLOTTER CMDS*
140 REM * TO INITIATE, AND   *
150 REM * TO SET VIEWPORT    *
160 REM * AND WINDOW         *
170 REM ************************
180 PRINT "IN;"
190 PRINT "IP 0,0,10100,7140;"
200 PRINT "SC 0,10,0,7.5;"
210 REM ************************
```

```
220 REM * SEND PLOTTER  CMDS*
230 REM * TO SELECT PEN,       *
240 REM * PLOT ABSOLUTE, &    *
250 REM * PLOT BORDER  W/PEN*
260 REM * UP AND PEN DOWN   *
270 REM ************************
280 PRINT "SP 1;"
290 PRINT "PA,PU 0,0 PD 9.8,0 9.8,7 0,7 0,0;"
300
310
  .
  .
  .
  .
  .

600 REM ************************
610 REM * RETURN PEN TO      *
620 REM * HOLDER & CEASE     *
630 REM * OUTPUT TO SLOT # 2 *
640 REM ************************
650 PRINT "SP 0;"
660 PRINT D$;"PR#0"
670 END
```

RUN, debug, and SAVE this program with a unique name (COG.010.2).

3. Read about the HP GL commands AA, AR, and CI. Modify the previous programs, COG. 010.1 or COG.010.2, to plot views of an object that contains circles and arcs. With your instructor's help, select such an object from a drafting text or workbook.

[COG.010.3]

```
10 REM **   ACTIVITY  TEN   **
20 REM * COMPUTER GRAPHICS *
30 REM ************************
40 REM * PLOTS  ARCS/CIRCLES *
50 REM * USING THE  HEWLETT *
60 REM * PACKARD  PLOTTER   *
70 REM ************************
80 REM * SELECT  SLOT  FOR     *
90 REM * PLOTTER  OUTPUT      *
95 REM ************************
100 LET D$ = CHR$ (4)
110 PRINT D$; "PR#2"
120 REM ************************
130 REM * SEND PLOTTER  CMDS*
140 REM * TO  INITIATE,  AND    *
150 REM * TO  SET  VIEWPORT    *
160 REM * AND  WINDOW          *
170 REM ************************
180 PRINT "IN;"
190 PRINT "IP 0,0,10100,7140;"
```

```
200 PRINT "SC 0,10,0,7.5;"
210 REM ************************
220 REM * SEND PLOTTER CMDS*
230 REM * TO SELECT PEN,       *
240 REM * PLOT ABSOLUTE, &    *
250 REM * PLOT BORDER W/PEN *
260 REM * UP AND PEN DOWN    *
270 REM ************************
280 PRINT "SP 1;"
290 PRINT "PA,PU 0,0 PD 9.8,0 9.8,7 0,7 0,0;"
300
310
 .
 .
 .
 .

600 REM ************************
610 REM * RETURN PEN TO       *
620 REM * HOLDER & CEASE     *
630 REM * OUTPUT TO SLOT # 2 *
640 REM ************************
650 PRINT "SP 0;"
660 PRINT D$;"PR#0"
670 END
```

RUN, debug, and SAVE this program with a unique name (COG.010.3).

4. For an optional activity, develop an interactive program that utilizes the plotter and the mouse. The program could allow the user to draw an object on the screen with the mouse and then with a suitable command, output the drawing to the plotter. Before you begin this project, you may want to review the use of variables with HP GL commands, and the use of arrays to store drawing data for later use.

Evaluation Criteria: To evaluate the student's success in completing this activity, the teacher should give the student a two part test. The first part should be a performance test where the student is required to write a high-resolution computer graphics program incorporating DOS and ProDOS commands, especially IN# and PR# and the HP GL mnemonic commands. The program should incorporate a Hewlett Packard pen plotter or other suitable output device. The student should be instructed to use only those commands covered in this activity or those covered in activities COG-004, COG-005, COG-006, COG-007, COG-008, and COG-009. The second part should be a recall test requiring the student to define the terms, viewport, window, absolute coordinates, relative coordinates, and "A", "B", and "C" size paper.

To be considered successful, the student's programs should run on the computer without error, allow output to the plotter, and plot the intended image. On the second part, the student should be able to provide acceptable definitions for the terms. They will be considered acceptable if they agree with definitions found in current computer literature.

Microcomputer Graphics Applications

COMPUTER GRAPHICS PAINTING

Introduction: Computer paint systems are interactive computer graphics systems used to create 2D computer images that simulate the characteristics of an image painted on canvas or paper. These systems provide the user with the computer equivalent of a canvas, paint brush, air brush (or spray can), pencil, straightedge, eraser, type, and much more. While these systems have limitations when compared to the conventional method of painting, they are less messy and in many ways, easier to experiment with.

Enabling Objective: Before beginning this activity, the student should have completed Activity COG-003, How to Operate the Apple Computer. He or she should be able to turn the computer off and on, be familiar with the keyboard, be able to load and save programs to the disk drive, and be able to use the printer. If students are to work through both this activity and COG-013, Computer Graphics Animation, or COG-014, Desktop Publishing, this activity should be done first.

Performance Objective (15-20 days):

TASK/COMPETENCY--COG-011

Upon completion of this activity, the student will be able to describe a computer paint system application and use a system to paint computer generated images. The student will also be able to recall definitions for the following:

palette	shade
drag	shadow
cut and paste	pixel
fill pattern	fat bits
fill pattern leak	basic colors of light

CONDITIONS

The above tasks will be performed under the following conditions:

a) The terms will be recalled from memory.

b) The student will be provided resource materials when she or he is asked to write a description of or describe orally an application.

c) To generate the painted image, the student should be provided a outline (drawn on paper) of the 2D image to be painted; an Apple IIe (with 128K) or the IIc computer system with at least one disk drive (2 are suggested); a mouse, Apple tablet, KoalaPad, or joystick; a color monitor; and the software package *Dazzle Draw*; or an Apple IIgs and either the software package *Paintworks* or *Dazzle Draw*.

STANDARDS OF PERFORMANCE

The above tasks should be performed to the following standards:

a) The student will be expected to provide a definition (similar to that found in the current computer graphics literature) for 3 out of 4 terms selected randomly.

b) The description of an application should include the advantages of using computer paint systems over conventional techniques, the types of computers used, careers associated with the application, and some specific examples of work accomplished with this application.

c) The 2D painted image will require the use of at least five paint system functions. The painted image will compare favorably with a painted image prepared by the instructor.

Selected Instructional Materials: An Apple IIe, IIc, or IIgs computer with printer; color monitor; disk drive (preferably two); and a mouse, Apple tablet, KoalaPad, or joystick; and the application program *Blazing Paddles, Dazzle Draw, DeluxePaint II*, or *Paintworks Plus*. Note: *DeluxePaint II* and *Paintworks Plus* will only run on an Apple IIgs. Also needed are the manuals or user's guides which are provided with the software packages.

Teacher References

Recommended

Books:	The user's guide or user's manual that is supplied with one of the software packages listed below.
Videotape:	Frost and Sullivan, Inc. (1987). *Graphic arts systems/video text report (VTR) 002*. A Frost and Sullivan video text report. Available from Frost and Sullivan, Department R-1, 106 Fulton Street, New York, NY 10038.
Software:	One of the following:
	Darooge, M., McCullough, E. & Walker, D. (198_). *Blazing Paddles*. Available from Baudville, 1001 Medical Park Drive S.E., Grand Rapids, MI 49506 or a local computer software dealer.
	Silva, D. (198_). *DeluxePaint II*. Available from Electronic Arts, 1820 Gateway Drive, San Mateo, CA 94404 or a local computer software dealer.
	Snider, D. (1984,1987). *Dazzle Draw*. Available from Broderbund Software, Inc., P.O. Box 12947, San Rafael, CA 94913-2947 or a local computer software dealer.

Version Soft, Inc. (1986). *Paintworks Plus*. Available from Activision, Inc., P.O. Box 7287, Mountain View, CA 94039 or a local Apple computer dealer.

Suggested

Books:

Kemnitzer, R.B. (1983). *Rendering with markers*. New York: Watson-Guptill Publications.

Martin, C.L. (1968). *Design graphics*. New York, NY: The Macmillan Company.

Ungar, J. (1985). *Rendering in mixed media*. New York, NY: Watson-Guptill Publications.

Teacher Activities

1. Familiarize yourself with computer paint systems by reading the guide that accompanies the paint program you have chosen for your students to use and by using the paint program.

2. Using your favorite drafting textbook, review the following and prepare demonstrations for your students. Although you can use other media (such as the blackboard) to demonstrate, use the computer when possible:

   Multiview drawings

   Shape description.
   Reproducing irregular shapes with grids

   Pictorial drawing

   Isometric drawings
   Perspective drawings

   Using your favorite rendering textbook, review the following and prepare demonstrations for your students. You should emphasize how the basic geometric shapes are rendered (cubes, cylinders, cones, spheres, and pyramids):

   Rendering techniques

   Shades in black and white
   Shadows in black and white
   Shades and Shadows in color

3. If time permits, work through the student activities.

4. Prepare for and demonstrate the software package you plan for your students to use.

5. Be prepared to offer suggestions for Part 8 of the student activities. For some students who have difficulty drawing and painting objects or determining what they should draw and paint, you may need to give them objects to paint (give them drawings similar to what you would find in a coloring book and let them paint them). However, this should be done only as a last resort.

6. When showing the videotape, *Graphic Arts Systems/Video Text Report (VTR) 002*, you should skip (fast forward over) those parts that are highly technical and those parts that deal with solid modeling. The parts of the video that are important to this activity are those parts that show 2D paint systems.

Student References

Recommended

Books:

The user's guide or user's manual that is supplied with one of the following software packages.

Videotape:

Frost and Sullivan, Inc. (1987). *Graphic arts systems/video text report (VTR) 002*. A Frost and Sullivan video text report. Available from Frost and Sullivan, Department R-1, 106 Fulton Street, New York, NY 10038.

Software:

One of the following:

Darooge, M., McCullough, E. & Walker, D. (198_). *Blazing Paddles*.

Silva, D. (198_). *DeluxePaint II*.

Snider, D. (1984,1987). *Dazzle Draw*.

Version Soft, Inc. (1986). *Paintworks Plus*.

Suggested

Books:

Kemnitzer, R.B. (1983). *Rendering with markers*. New York: Watson-Guptill Publications.

Martin, C.L. (1968). *Design graphics*. New York, NY: The Macmillan Company.

Ungar, J. (1985). *Rendering in mixed media*. New York, NY: Watson-Guptill Publications.

Student Activities

1. Observe your instructor as he or she demonstrates the techniques for representing objects with multiview, isometric, and perspective sketches and drawings.

2. Try using these techniques to sketch simple objects with paper and pencil.

3. As the instructor makes future demonstrations and when you view the videotape in Part 4 below, observe how the following terms are used:

palette	shade
drag	shadow
cut and paste	pixel
fill pattern	fat bits
fill pattern leak	basic colors of light

   If you do not understand any of these terms, be sure and ask your instructor for a definition.

4. View those parts of the videotape, *Graphic Arts Systems/Video Text Report (VTR) 002*, that deal with 2D paint systems.

5. Observe your instructor as he or she demonstrates how objects are made to look three dimensional by using shades and shadows.

6. Try these techniques by sketching spheres, cubes, pyramids, cylinders, and cones with surfaces shaded and shadows plotted. Sketch the objects with paper and pencil. Next, try to sketch the same objects with shades and shadows drawn in color with magic marker or colored pencil.

7. Observe your instructor as he or she demonstrates the use of a computer paint system.

8. Practice using the paint program:

   a) Read the user's guide.

   b) Practice using the different functions.

   c) When you think you are ready, use the paint program to draw and paint a simple flat object or design. The following are offered as suggestions:

      (1) a vanity license plate for an automobile or a bicycle.

      (2) a personal "T" shirt design.

      (3) a school sports emblem for a football helmet, baseball cap, or "T" shirt.

(4) a design for the cover of a wall poster, notebook, school play program, sports event program, or school newspaper masthead.

(5) a logo for your personal calling card or one for a parent.

(6) a logo for a fictitious company.

d) Next, use the paint program to draw and paint a single multiview of an object, such as:

(1) a front view of a small house or a business entrance.

(2) the side view of car.

(3) a front view of your computer.

(4) a top view of the family house and lot with future landscaping designs.

(5) a front or a side view of a piece of furniture.

With the above multiviews, you can begin to incorporate shadows and shading to show some depth, for example, show shadows around doors and windows in the front view of the house.

e) Use the paint program to draw and paint the pictorial drawings of the geometric shapes you drew in Part 5 above.

f) Lastly, use the paint program to draw and paint objects and scenes using isometrics and perspectives. Use shades and shadows to help show depth and project realism. Following are some ideas for this exercise:

(1) an isometric or perspective of a model car design.

(2) a perspective of a small house designed by you.

(3) a new design for the interior of your bedroom.

(4) an isometric of a machine part, tool, or other object found in your technology education laboratory.

9. As an optional activity, create a painting with your paint system and enter it in one of the many computer art contests for high school students. Magazines such as *A+* and *Classroom Computer Learning* announce contests such as this from time to time.

Evaluation Criteria: The student should be given a short quiz where she or he is to define four randomly selected terms from this unit. Her or his definitions should conform to definitions found in the recommended or suggested texts or videotapes. She or he should be able to define three of four randomly selected terms successfully.

The student should also be evaluated on how well she or he can describe a computer paint application. The student's description of an application should include the advantages of using computer painting over conventional techniques, the types of computers used, careers associated with the application, and some specific examples of work accomplished with this application. Students should be given extra credit if their examples are not like those described in the videotape or the text (required additional research), and if they demonstrate an exceptional amount of enthusiasm in their presentation.

The third and last part of the evaluation should be based on the student's creativity, technique, and enthusiasm in painting images with the paint program. Special attention should be given to Student Activity, Part 8. High marks should be given for creativity when the student is able to develop a number of different ideas for paintings, when the paintings are unlike those of other students in the class, and when the paintings contain elaborate detail. High marks for technique should be given when the student has correctly plotted shadows, surfaces are correctly shaded, and highlighting enhances the 3D effect. Extra credit should also be given for scenes involving human figures or parts of the human anatomy.

Microcomputer Graphics Applications
COMPUTER AIDED DESIGN

Introduction: Computer aided design (CAD) systems are interactive computer graphics systems used to create 2D or 3D drawings of topographical maps, mechanical or electrical devices, architectural structures, and other entities designed or altered by engineers, architects, and designers. This computer graphics application has drastically changed the way that we design items and the way they are graphically represented.

Enabling Objective: Before beginning this activity, the student should have completed Activity COG-003, How to Operate the Apple Computer. He or she should be able to turn the computer off and on, be familiar with the keyboard, be able to load and save programs to the disk drive, and be able to use the printer.

Performance Objective (15-20 days):

TASK/COMPETENCY--COG-012

Upon completion of this activity, the student should be able to describe a 2-D computer aided drafting system application and use a system to produce technical drawings. He or she should also be able to recall definitions for the following:

CAD	wireframe
digitizing	window
grid	viewport
resolution	Cartesian coordinates
origin	isometric drawings
orthographic drawings	relative coordinate
layers	perspective drawings
polyline	zoom
mirror	cursor
CAD function	jaggies
clipping	input
output	interactive
AEC	CIM
IGES	absolute coordinate

CONDITIONS

The above tasks will be performed under the following conditions:

a) The terms will be recalled from memory.

b) The student will be provided resource materials when she/he is asked to write a description of or orally describe an application.

c) The student will be provided the object they are to make a drawing of or a complete graphical description. An acceptable practice is to provide them a drawing in a different form than that which is expected (e.g., an isometric when a multiview is expected). The student will also be provided an Apple IIe, IIc, or IIgs computer system with at least one disk drive (2 are suggested); a mouse, Apple tablet, KoalaPad, or joystick; a monitor; and a CAD software package.

STANDARDS OF PERFORMANCE

The above tasks should be performed to the following standards:

a) The student will be expected to provide a definition (similar to that found in the current computer graphics literature) for 7 out of 10 randomly selected terms.

b) The description of an application should include the advantages of using computer aided drafting (CAD) systems over conventional techniques, the types of computers used, careers associated with the application, and some specific examples of work accomplished with this application.

c) The drawing will require the use of at least five CAD functions. The drawing will be correct and conform to standard drafting conventions as described in the ANSI drafting standards.

Selected Instructional Materials: An Apple IIe, IIc or IIgs computer with printer; monitor; disk drive (preferably two); and a mouse, Apple tablet, KoalaPad, or joystick; a CAD application program; and the documentation provided with the application program.

Teacher References

Recommended

Books:

The user's guide or user's manual that is supplied with the software package that has been selected for use in this activity, and one of the following:

Bertoline, G.R. (1985). *Fundamentals of CAD*. Albany, NY: Delmar Publishing, Co.

Goetsch, D.L. (1985). *Computer-aided drafting*. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Voisinet, D.D. (1986). *Introduction to computer-aided drafting (2nd edition)*. New York, NY: Gregg Division of McGraw-Hill.

Videotapes:

CIM Solutions. (1986). *Computer aided design: a video evaluative report I*. Available from CIM Solutions, University Station, P.O. Box 7041, Provo Utah 84602.

CIM Solutions. (1986). *Computer aided design: a video evaluative report II*. Available from CIM Solutions, University Station, P.O. Box 7041, Provo Utah 84602.

Frost and Sullivan, Inc. (1987). *Workstations/video text report (VTR) 001*. A Frost and Sullivan video text report. Available from Frost and Sullivan, Department R-1, 106 Fulton Street, New York, NY 10038.

Silicon Graphics Computer Systems. (1987). *Superworkstation demonstration*. Silicon Graphics, Corporate Headquarters, 2011 Suerlin Road, Mountain View, CA 94043.

Software:

One of the following:

Langer, V. (1985). *MATC-CAD* Available from the Milwaukee Area Technical College, 1015 North Sixth Street, Milwaukee, Wisconsin 53203.

Versacad Corporation. (1987). *Entry Level CADApple*. Available from Versacad Corporation, 7372 Prince Drive, Huntington Beach, CA 92647 or a local computer software dealer.

Suggested

Books:

Lamit, L. G., & Paige, V. (1987). *Computer-aided design and drafting*. Columbus, OH: Merrill.

McKissick, M. L. (1987). *Computer-aided drafting and design*. Englewood Cliffs, NJ: Prentice-Hall.

Voisinet, D.D. (1987). *Computer-aided drafting and design, concepts and applications*. New York, NY: Gregg Division of McGraw-Hill.

Teacher Activities

1. Familiarize yourself with computer aided design systems and terminology by reading the text you selected from the recommended list above or one of the suggested texts.

2. Preview the videotapes carefully. You may find that they are not suitable for your students. This is especially true of the Frost and Sullivan tape, *Workstations/Video Text Report (VTR) 001*. It is a very technical tape on state-of-the-art workstations. If you do show it to your students, be sure to discuss terminology with them beforehand. The two CIM Solutions tapes should not be shown in their entirety. You should select the demonstration sequence for only one or two of the systems. The Silicon Graphics is a short tape and is probably the best for showing future directions in CAD.

3. Familiarize yourself with the CAD software you have selected for this activity.

4. If time permits, work through the student activities.

5. Prepare to demonstrate the CAD package you plan for your students to use.

6. Select drafting problems for the students to do in Part 4 of the Student Activities. The following will help guide you in making those selections:

For:	Type of problem suggested:
4,b	Figures 6-42 through 6-47 in *Fundamentals of CAD* by Bertoline. Select a problem like one of these and have them draw a multiview without dimensions.
4,c),(1)	Figures 6-38 or 6-39 in *Fundamentals of CAD* by Bertoline. Select a problem like one of these and have the students draw an isometric without dimensions.
4,c),(2)	Figures 6-34 or 8-36 in *Fundamentals of CAD* by Bertoline. Select a problem like one of these and have the students draw a fully dimensioned multiview.
4,c),(3)	Figures 9-38, 9-40, and 9-42 in *Fundamentals of CAD* by Bertoline. Select a problem like these. Have the students draw a set of electrical symbols similar to those in Figure 9-38 and then use them to draw a schematic similar to Figure 9-40 or 9-42.
4,c),(4)	Figure 9-39. Have students draw architectural symbols similar to those in this figure and then use them to draw a simple architectural structure of their own design.

7. Plan a field trip to a local engineer's, or architect's office or to a local business or industry that uses a CAD system. Refer to Acitivity COG-002.

Student References

Recommended

Books:

The user's guide or user's manual that is supplied with the software package that has been selected for use in this activity, and one of the following:

Bertoline, G.R. (1985). *Fundamentals of CAD*. Albany, NY: Delmar Publishing, Co.

Goetsch, D.L. (1985). *Computer-aided drafting*. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Voisinet, D.D. (1986). *Introduction to computer-aided drafting (2nd edition)*. New York, NY: Gregg Division of McGraw-Hill.

Videotapes:

CIM Solutions. (1986). *Computer aided design: a video evaluative report I*.

CIM Solutions. (1986). *Computer aided design: a video evaluative report II*.

Frost and Sullivan, Inc. (1987). *Workstations/video text report (VTR) 001*.

Software:

One of the following:

Langer, V. (1985). *MATC-CAD*.

Versacad Corporation. (1987). *Entry Level CADApple*.

Suggested

Books:

Lamit, L. G., & Paige, V. (1987). *Computer-aided design and drafting*. Columbus, OH: Merrill.

McKissick, M. L. (1987). *Computer-aided drafting and design*. Englewood Cliffs, NJ: Prentice-Hall.

Voisinet, D.D. (1987). *Computer-aided drafting and design, concepts and applications*. New York, NY: Gregg Division of McGraw-Hill.

Student Activities

1. Read the introductory chapters in the textbook your instructor has selected from the list of recommended books shown above. When you have completed your reading, you should be able to recall the definitions for each of the following terms:

CAD	wireframe
digitizing	window
grid	viewport
resolution	Cartesian coordinates
origin	isometric drawings
menu	orthographic drawings
layers	perspective drawings
polyline	zoom
mirror	cursor
CAD function	jaggies
clipping	input
output	interactive
AEC	CIM
IGES	absolute coordinates
relative coordinates	

2. If your instructor recommends it, view those portions of the videotapes she or he selects.

3. Observe your instructor as she or he demonstrates the computer aided design (CAD) application program that has been selected for your use in this activity.

4. Practice using the CAD application program:

   a) Read the user's guide, reference manual, or student workbook that accompanies the software package.

   b) Use the CAD software package to produce a multiview (2 or 3 view) drawing of a simple object without dimensions. Your instructor will supply you with a problem.

   c) Continue to learn more about the CAD software by producing the following drawings:

      (1) an isometric drawing of a simple object.

      (2) a multiview drawing of a simple object fully dimensioned.

      (3) a schematic drawing of a simple electrical circuit.

      (4) a floor plan a of simple architectural structure (something similar to a child's playhouse).

7. Prepare a list of questions to ask those CAD professionals you will visit on the field trip that your instructor has planned for you and your class. You may want to learn more about their professional training, the types of drawings they use their CAD system for, the type of system they use, etc.

8. After returning from your field trip, write a short paper explaining how CAD systems are used in industry. How did the professionals you visited use CAD systems to increase productivity? What kinds of output do they generate? How are CAD professionals trained? What are some of the problems encountered when implementing CAD into an operation?

Evaluation Criteria: The student should be given a short quiz where she/he is to define ten randomly selected terms from this unit. Her/his definitions should conform to definitions found in the recommended or suggested texts or videotapes. She/he should be able to define seven of the ten terms successfully.

The student should also be evaluated on how well she/he can describe a computer aided design application (see Student Activities, Part 8). The student's description of an application should include the advantages of using computer aided design over conventional techniques, the types of computers used, careers associated with the application, and some specific examples of work accomplished with this application. Students should be given extra credit if their examples are not like those described in the videotape, the text, or on the field trip (it required additional research), and if they demonstrate an exceptional amount of enthusiasm in their presentation.

The third and last part of the evaluation should be based on the student's skill and enthusiasm in generating drawings with the CAD application. Special attention should be given to Student Activity, Part 4, c). The drawings generated should be correctly drawn and dimensioned (when dimensioning is called for) according to ANSI drafting standards. The student should also demonstrate a certain amount of speed and the ability to work without help from the teacher or other students.

Microcomputer Graphics Applications
COMPUTER AIDED ANIMATION

Introduction: Animation, the art of making drawings that appear to move, used to require expensive movie making equipment and loads of artistic talent. While artistic talent is still required, computers have simplified the task tremendously. Now, most anyone with a personal computer, can do simple animation. Although the animation created on a personal computer is not as eloquent as that which is produced in the Walt Disney Studio or with large super computers, it does approach that which is used in arcade game animation.

Computer generated animation is finding applications in many new fields of endeavor. The computer is being used to generate animation sequences in the traditional areas of movie making, cartoons for television, arcade games, and television commercials. New applications are also being found in flight and space simulation, education, medical research, engineering, sports, chemistry, and biology. In most of these applications, computer animation is used to simulate the physical laws of nature. It is now much more practical for these new applications, because the computer has taken over much of the laborious task of drawing hundreds of animation cells (24-30 per second of animation) needed to create a few minutes of film or videotape.

Enabling Objective: Before beginning this activity, the student should have completed Activity COG-003, How to Operate the Apple Computer. He or she should be able to turn the computer off and on, be familiar with the keyboard, be able to load and save programs to the disk drive, and be able to use the printer. If students are to work through both this activity and Activity COG-011, Computer Graphics Painting, Activity COG-011 should be done first.

Performance Objective (15-20 days):

TASK/COMPETENCY--COG-013

Upon completion of this activity, the student will be able to describe a computer generated animation application, develop a storyboard, and use the storyboard and a computer system to produce a 2D animation sequence. The student should also be able to:

flicker
pixel
object blend sequence
flip book
storyboard
persistence of vision
frame buffer
real time animation
thaumatrope

CONDITIONS

The above tasks will be performed under the following conditions:

a) The animation terms will be recalled from memory.

b) The student will be provided resource materials when she or he is asked to write a description of or describe orally an application.

c) To generate the animation sequence, the student should be provided a series of animation cells drawn on paper and an Apple IIe (with 128K) or the IIc computer system with at least one disk drive (2 are suggested); a mouse, Apple tablet, KoalaPad, or joystick; and the software package *Animate*; or an Apple IIgs and the software package *Paintworks Plus* or *Panitworks Gold.*

STANDARDS OF PERFORMANCE

The above tasks should be performed to the following standards:

a) The student will be expected to provide a definition (similar to that found in the current computer graphics literature) for 3 out of 4 terms selected randomly.

b) The description of an application should include the advantages of using computer animation over conventional techniques, the types of computers used, careers associated with the application, and some specific examples of work accomplished with this application.

c) The animated sequence will contain at least six cells and represent a smooth fluid movement of the image. The storyboard and individual cells will be skillfully drawn (will compare favorably to an example prepared by the instructor).

Selected Instructional Materials: An Apple IIe or IIc computer with printer; monitor (preferably with color); disk drive (preferably two); and a mouse, Apple tablet, KoalaPad, or joystick; and the application program *Animate*; or the Apple IIgs and *Paintworks Plus or Paintworks Gold.* Also needed are the references *Animation, The Animation Book,* and *Computer Animation Primer*.

Teacher References

Recommended

Books:

Fox, D., & Waite, M. (1984). *Computer animation primer*. New York: McGraw-Hill.

Blair, P. (1986). *Animation*. Tustin, CA: Foster Art Service Inc.

Laybourne, K. (1979). *The animation book*. New York: Crown.

Paintworks plus user's guide or *Animate user's manual*. These are supplied with the software.

Videotapes:

Dupont, C. & Cinemagic Productions. (1987). *Computer magic*.

Electronic Pictures Corporation. (1987). *Computer graphic special on TV commercials and animation.*

National Computer Graphics Association. (1988). *1988 International computer animation competition winners*.

Software: *Paintworks plus* by Activision (Apple IIgs only) or *Animate* by Broderbund Software.

Teacher Activities

1. Familiarize yourself with computer animation by reading the following:

   a) Part One of *Computer Animation Primer* by Fox and Waite.

   b) *Animation* by Preston Blair.

   c) *The Animation Book* by Kit Laybourne.

2. Preview the videotapes *Computer Magic, Computer Graphic Special on TV Commercials and Animation,* and *1988 International Computer Animation Competition Winners*.

3. Familiarize yourself with the software package *Animate* or *Paintworks Plus*. If you have a IIgs, *Paintworks Plus* is suggested.

4. If time permits, work through the student activities.

5. Prepare a demonstration to demonstrate the software package you plan for your students to use.

6. Be prepared to offer suggestions for Part 10 of the student activities. For those students who have difficulty with this part, be prepared to provide a lot of encouragement and as a last resort, a sequence that they can copy.

Student References

Recommended

Books: Fox, D., & Waite, M. (1984). *Computer animation primer*. New York: McGraw-Hill.

Blair, P. (1986). *Animation*. Tustin, CA: Foster Art Service Inc.

Paintworks plus user's guide or *Animate user's manual*. These are supplied with the software.

Videotapes: Dupont, C. & Cinemagic Productions. (1987). *Computer magic*.

Electronic Pictures Corporation. (1987). *Computer graphic special on TV commercials and animation.*

National Computer Graphics Association. (1988). *1988 International computer animation competition winners.*

Software: *Paintworks plus* by Activision (Apple IIgs only) or *Animate* by Broderbund Software.

Suggested

Books: Laybourne, K. (1979). *The animation book.* New York: Crown.

Student Activities

1. View the videotape, *Computer Magic.*

2. Read Chapter One in *Computer Animation Primer.* As you read, make a list of new terms that are associated with computer animation. When you are through, you should be able to define the terms:

flicker	persistence of vision
pixel	frame buffer
object blend sequence	real time animation
flip book	thaumatrope
storyboard	

3. Scan through the book *Animation.* Try sketching some of the figures found in this book. If it is available, scan through *The Animation Book.*

4. View the videotape, *Computer Graphic Special on TV Commercials and Animation.*

5. Observe your instructor as she or he demonstrates the *Animate* or *Paintworks Plus* application program.

6. Practice using either the *Animate* or *Paintworks Plus* application program:

   a) Read the user's guide that accompanies the software package.

   b) Use the software package to design a simple animation sequence. A suggested sequence is that of a bouncing ball.

      (1) Sketch a five cell storyboard. Draw a ball in four suspended positions, one above the other and equally spaced, and for the fifth cell, draw the ball colliding with the floor.

      (2) Now draw the five cells on the computer. Hint: use the zoom and copy features of the software. Make the ball have a three dimensional effect by adding highlights and shading. Realism can also be enhanced by adding a shadow on the floor. You may even want to draw background scenery--a basketball or volleyball court.

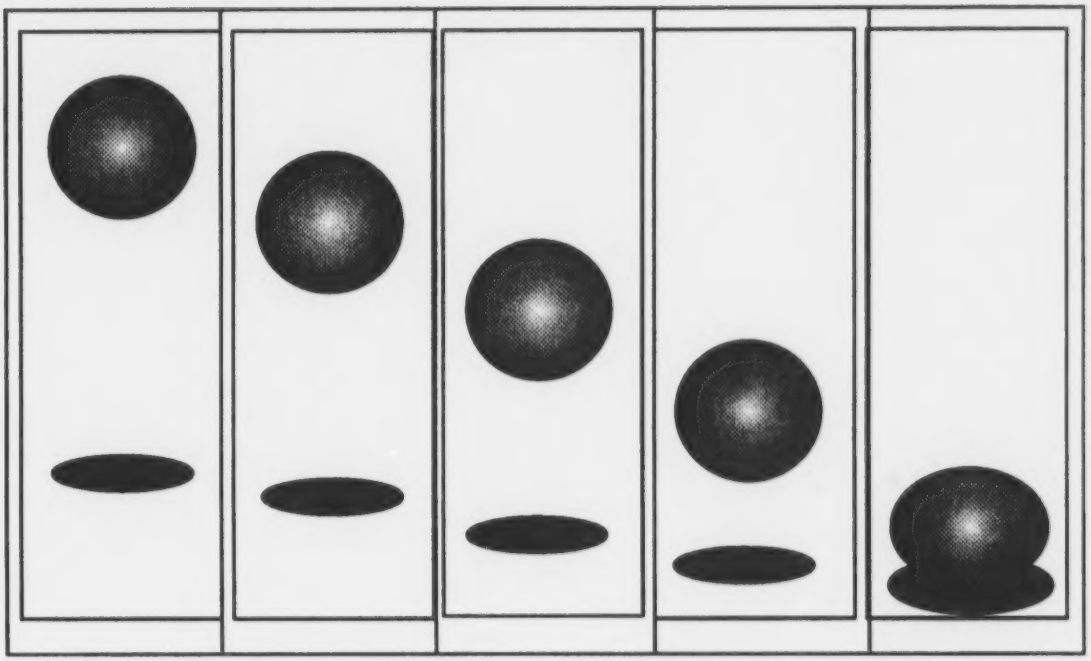


Figure 12.--Five Animation Cells of a Bouncing Ball

(3) When you have finished drawing all of the cells, animate the sequence by stepping through the cells from the ball's highest position to the collision cell and then back through the cells in reverse order to the highest position. When finished, you should have a nine cell animation sequence.

(4) Be sure to save your animation sequence on a diskette. If the proper type of video recording equipment is available, record your sequence on videotape.

7. View the videotape, *Computer Graphic Special on TV Commercials and Animation*. When you finish, write a short report on "Computer Generated Animation." In the report, give a brief description of how it is used (give specific examples), the types of computers used, the types of careers associated with this field, and tell how computers have benefitted the art of animation. When doing the research for this report be sure to consult with your librarian for additional books on the subject. You may also want to write your report using a computer and a word processing program.

8. Animating a ball and animating a human or animal figure require different techniques. Using the software you used to animate the ball, draw the eight cells shown below of the human stick figure. After you have the cells drawn, animate the figure. Next, add your own background. If your software permits, add sound. Be sure to save your sequence on disk.

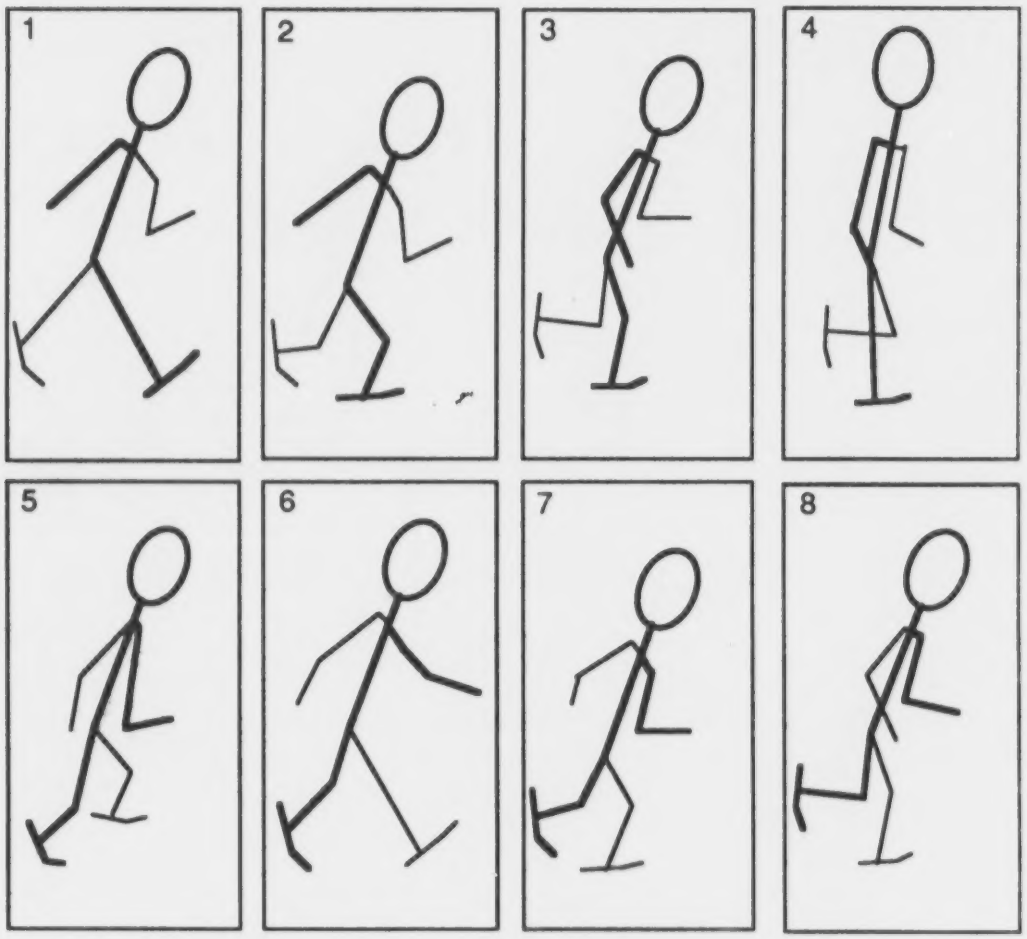


Figure 13--Eight Animation Cells of a Walking Stick Figure

9. View the videotape, *1988 International Computer Animation Competition Winners*. Afterward, make a list of all the animation sequences you would like to make. Try to make the list as long as you can and try not to list any that you viewed on the tape. Try to be as specific as you can, for example: "a short sequence of a boy playing with a yo-yo" would be preferred over "a cartoon for Saturday T.V."

10. From the above list, select an animation sequence for your next project. Go back and scan through the book, *Animation*, again. Look for ideas for your project. Develop the storyboard. With paper and pencil, make preliminary sketches of each of the cells. At first, try to make it simple. You can elaborate later. Before you begin to work with the computer, check with your instructor, for her/his suggestions.

11. If the equipment is available, record your sequence on videotape. Consider entering your project in an animation competition.

Evaluation Criteria: The student should be given a short quiz where she or he is to define four randomly selected terms from this unit. Her/his definitions should conform to definitions found in the recommended or suggested texts or videotapes. She or he should be able to define three of the four terms successfully.

The student should also be evaluated on how well she or he can describe a computer generated animation application (see Student Activities, Part 7). The student's description of an application should include the advantages of using computer animation over conventional techniques, the types of computers used, careers associated with the application, and some specific examples of work accomplished with this application. Students should be given extra credit if their examples are not like those described in the videotape or the text (required additional research), and if they demonstrate an exceptional amount of enthusiasm in their presentation.

The third and last part of the evaluation should be based on the student's skill and enthusiasm in developing the animation sequences. Special attention should be given to Student Activity, Parts 9 and 10. The animated sequence should contain at least six cells and represent a smooth fluid movement of the image. The storyboard and individual cells will be skillfully drawn (will compare favorably to an example prepared by the instructor). Students should be given extra credit for: character animation, the movement of humans or animals, as opposed to a 2D object sliding across the screen; 3D effects, such as shadows, shading, and highlighting; and added features such as backgrounds, sound, etc.

Microcomputer Graphics Applications
DESKTOP PUBLISHING

Introduction: Desktop publishing systems are interactive computer systems that perform page or document processing. Text and graphics are manipulated on a page or pages to create reports, newsletters, books, pamphlets, and fliers. These systems use personal computers, scanners, and printers (usually laser printers) to perform those tasks necessary to publish a document--create the original material, layout pages, and print the final camera-ready copy. As with many other computer graphics applications, desktop publishing is revolutionizing the communications industry.

Enabling Objective: Before beginning this activity, the student should have completed Activity COG-003, How to Operate the Apple Computer. He or she should be able to turn the computer off and on, be familiar with the keyboard, be able to load and save programs to the disk drive, and be able to use the printer. It would also be desirable to have the student complete Activity COG-011, Computer Graphics Painting, before beginning this activity.

Performance Objective (15-20 days):

TASK/COMPETENCY--COG-014

Upon completion of this activity, the student should be able to describe a publishing system application and to use a system to produce camera ready copy for publication. The student should also be able to recall definitions for the following:
:

text fonts	greeking
justification	leading
binding	letterspacing
photocopying	snap-to
offset printing	kerning
proofreading	optical character readers
WYSIWYG	scanners

CONDITIONS

The above tasks should be performed under the following conditions:

a) The terms should be recalled from memory.

b) The student should be provided resource materials when she or he is asked to write a description of or orally describe an application.

c) To generate the camera ready copy, the student should be provided with rough drafts of articles and news items; an Apple IIe, IIc, or IIgs computer system with at least one disk drive (2 are suggested); appropriate input devices; a color monitor; and a desktop publishing software application package.

STANDARDS OF PERFORMANCE

The above tasks should be performed to the following standards:

a) The student should be expected to provide a definition (similar to that found in the current computer graphics literature) for six out of eight terms selected randomly.

b) The description of an application should include the advantages of using desktop publishing systems over conventional techniques, the types of computers used, careers associated with the application, and some specific examples of work accomplished with this application.

c) The camera ready copy should require the use of at least five of the desktop publishing system functions. The camera ready copy should be free of spelling, hyphenation, and grammatical errors. It should also conform with page layout design principles.

Selected Instructional Materials: An Apple IIe (may need to have ROM enhancement), IIc, or IIgs computer with 64K or 128K of RAM memory and a matrix printer; monitor; disk drive (preferably two); and the desktop publishing application program *Personal Newsletter, Publish It!, Springboard Publisher, Medley,* or *The Newsroom.* Also needed are the manuals that are included with the chosen software. Some of the above software requires 128K of RAM memory and some may be used with laser printers, as well as matrix printers. Matrix printers that are considered near letter quality (LQ) or laser printers are preferred for this activity. Other devices, especially a scanner and a photocopy machine or offset press, would also be very useful.

Teacher References

Recommended

Books:

The user's guide or user's manual that is supplied with the software package or packages chosen for this activity.

McNeil, D. (1987). *Compute!'s quick and easy guide to desktop publishing*. Greensboro, NC: Compute! Publications Inc.

Parker, R.C. (1988). *Looking good in print.*. Chapel Hill, NC: Ventana Press.

Software:

One of the following page oriented document and newsletter processors:

Milliken Publishing Co. (1988). *Medley*. Available form the Milliken Publishing Co., 1100 Research Blvd., St. Louis, MO 63132-0579 or a local computer software dealer.

Softsync, Inc. (1987). *Personal newsletter*. Available from Softsync, Inc., 162 Madison Ave., New York, NY 10016 or a local computer software dealer.

Springboard Software. (1988). *Springboard publisher*. Available from Springboard Software, 7808 Creekridge Circle, Minneapolis, MN 55435 or a local computer software dealer.

Springboard Software. (1985). *The newsroom.*. Available from Springboard Software, 7808 Creekridge Circle, Minneapolis, MN 55435 or a local computer software dealer.

Timeworks. (1988). *Publish it!* Available from Timeworks, 444 Lake Cook Road, Deerfield, IL 60015 or a local computer software dealer.

Suggested Software:

A word processor similar to one of the following or a paint program similar to those used in COG-013 may be used in conjunction with one of the above document processors. Check to see if the document processor you have selected will accept files from the word processor or the paint program, before you make a purchase:

Lissner, R.J. & Apple Computer, Inc. (1983-86). *AppleWorks*. Available from the Claris Corporation, 20525 Mariani Avenue, Cupertino, CA 95014 or a local computer software dealer.

Scholastic, Inc. (1987). *Bank street writer III*. Available from Scholastic, Inc., 730 Broadway, New York, NY 10003 or a local computer software dealer.

StyleWare. (1987,1988). *Multiscribe*. or *Multiscribe GS*. Available from StyleWare, 5250 Gulfton, Suite 2E, Houston, TX 77081 or a local computer software dealer.

Software similar to the following is available for special publishing applications such as designing banners, greeting cards, certificates, etc:

Balsam, D. & Kahn, M. (1984). *The print shop* or *Print shop GS*. Available from Broderbund Software, 17 Paul Drive, San Rafael, CA 94903-2101.

Baudville. (1986). *Award maker plus*. Available from Baudville, 5380 52nd Street S.E., Grand Rapids, MI 49506.

Springboard Software. (1986). *Certificate maker*. Available from Springboard Software, 7808 Creekridge Circle, Minneapolis, MN 55435

Bantam. (1987). *Walt Disney card and party shop*. Available from Bantam, 666 Fifth Avenue, New York, NY 10103.

Some of the desktop publishing software comes with clip art files or makes these files available to you for an additional price. Sometimes compatible files are available through third parties. You may find these to be very helpful.

Software similar to the following can also be useful in this activity if your students need help with writing articles, spelling or typing:

Super Scoop
Helpful Writer
Persuasive Essay
The Sensible Speller
Junior Typer
Typing Tudor IV
Mastertype

The above software is available from several sources; however, one source that has all of them plus other software relative to teaching writing, reading, spelling, and grammar is Opportunities for Learning, Inc., 20417 Nordhoff St., Department V A 3, Chatsworth, CA 91311. Refer to their catalog, *The Computer Approach to English*.

Periodicals: *Publish*. PCW Communications Inc., 501 Second Street, San Francisco, CA 94107. Subscription Information available from Subscriber Services, P.O. Box 55400, Boulder, CO 80322. A monthly publication.

Videotapes:

Teacher Activities

1. Familiarize yourself with desktop publishing by reading *Compute!'s Quick and Easy Guide to Desktop Publishing*. written by McNeil. Chapters 1, 2, 3, 7, 9, and 11 are especially important. Although this publication concentrates on desktop publishing with a Macintosh or IBM PC computer, keep in mind that the principles, concepts, and terms are alike regardless of the computer used. The main difference in desktop publishing on a Macintosh or IBM PC and desktop publishing on an Apple IIe, IIc, or IIgs is the quality of the output. With a laser printer as the output device on an Apple IIgs system and the better IIgs desktop publishing software, there is very little difference. Also read Section 1, and scan the remainder of Parker's *Looking Good in Print*.

2. Familiarize yourself with the software that you plan to use in this activity--desktop publishing, word processor, graphics paint systems, clip art files, and other application programs or files.

4. If time permits, work through the student activities.

5. Prepare to demonstrate the software packages you plan for your students to use.

6. Be prepared to offer suggestions for the student desktop publishing activities. This may include developing cooperative projects with other classes (journalism, English, etc.) student clubs, local civic organizations, or the school administration. Cooperative arrangements could have your students publishing materials for these classes, clubs, organizations, etc. Caution should be taken, however, to prevent the demands and deadlines of these other parties from dictating your curriculum.

Student References

Recommended

Books:

The user's guide or user's manual that is supplied with the software package or packages chosen for this activity.

McNeil, D. (1987). *Compute!'s quick and easy guide to desktop publishing*. Greensboro, NC: Compute! Publications Inc.

Parker, R.C. (1988). *Looking good in print*. Chapel Hill, NC: Ventana Press.

Software:

One of the following page oriented document and newsletter processors:

Milliken Publishing Co. (1988). *Medley*.

Softsync, Inc. (1987). *Personal newsletter*.

Springboard Software. (1988). *Springboard publisher*.

Springboard Software. (1985). *The newsroom*..

Timeworks. (1988). *Publish it!*

Suggested

Software:

Software similar to the following is available for special publishing applications such as designing banners, greeting cards, certificates, etc:

Balsam, D. & Kahn, M. (1984). *The print shop or Print shop GS*.

Baudville. (1986). *Award maker plus*.

Springboard Software. (1986). *Certificate maker.*

Bantam. (1987). *Walt Disney card and party shop.*

A word processor similar to one of the following or a paint program similar to those used in COG-013 may be used in conjunction with one of the above document processors.

Lissner, R.J. & Apple Computer, Inc. (1983,1986). *AppleWorks.*

Broderbund Software, Inc. (1987). *Bank street writer III.*

StyleWare. (1987,1988). *Multiscribe. or Multiscribe GS.*

Some of the desktop publishing software comes with clip art files or makes these files available to you for an additional price. Sometimes compatible files are available through third parties. You may find these to be very helpful.

Software similar to the following can also be useful in this activity:

Super Scoop
Helpful Writer
Persuasive Essay
The Sensible Speller
Junior Typer
Typing Tudor IV
Mastertype

Periodicals:

Publish. PCW Communications Inc., 501 Second Street, San Francisco, CA 94107.

Student Activities

1. Read Part 1 of and scan other parts of *Compute!'s Quick and Easy Guide to Desktop Publishing*. written by McNeil and Section 1 of Parker's *Looking Good in Print*. As you read, make a list of new terms that are associated with desktop publishing. When you are through, you should be able to define the following terms. If you don"t understand some of them, discuss them with your teacher. Rather than do all of this reading at one time, mix it in with some of the subsequent activities

text fonts	greeking
justification	leading
binding	letterspacing
photocopying	snap-to
offset printing	kerning
proofreading	optical character readers
WYSIWYG	scanners

2. Observe your instructor as she or he demonstrates the desktop application software that has been selected for you to use in this activity.

3. Practice using the desktop publishing application program your instructor has selected for this activity:

   a) Read the user's guide that accompanies the software package.

   b) Use the software package to design one of the following:

      (1) a greeting card.
      (2) a poster for the bulletin board announcing a:
          club meeting,
          a ball game,
          special event such as a cake sale, or
          an item for sale.
      (3) a business card.
      (4) letterhead stationery.
      (5) a schedule for a ball club.
      (6) a poster for a candidate in a class or student council election.
      (7) the school calendar for next month.
      (8) a teacher's daily schedule.

   Try to be original in your design. Remember, you can incorporate both text and graphics.

4. Visit a local print shop, newspaper, industry, or office that uses computers to publish materials. While on your visit, determine what software and hardware they use, the training required of the people involved in the publication, and the types of publications they generate on their system. It might also be interesting to find out if and how they published these materials before they acquired the desktop publishing system. Has the system improved productivity?

5. With your instructor's help, organize your class or group into a simulated publication organization and publish a newsletter, newspaper, brochure, or other publication. Your organization may need to have a publisher, editor, artists, writers, reporters, proof readers, press operators, and computer operators. More than likely, you and your classmates will want to perform one or more of these jobs by rotating through them.

Evaluation Criteria: The student should be given a short quiz where she or he is to define eight randomly selected terms from this unit. Her/his definitions should conform to definitions found in the recommended text. She or he should be able to define six of the eight terms successfully.

The student should also be evaluated on how well she or he can describe a computer desktop publishing application. The student's description of an application should include the advantages of using desktop publishing over conventional techniques, the types of computers used, careers associated with the application, and some specific examples of work accomplished with this application.

The third and last part of the evaluation should be based on the student's skill and enthusiasm in developing the publications. Special attention should be given to Student Activity, Parts 3 and 4. The publications should have required camera ready copy that used at least five desktop publishing system functions to produce. The publications should also be free of spelling, hyphenation, and grammatical errors. They should also conform with page layout design principles.

COMPUTER GRAPHICS RESOURCE LIST

Books

Allan, J. J. (1979). *The CAD/CAM glossary*. Dallas, TX: Productivity International, Inc.

A comprehensive glossary of computer-aided-design/computer-aided-manufacturing terms.

Apple Computer, Inc. (1983). *Apple II DOS user's manual*. Cupertino, CA: Author.

An Apple reference manual describing the Apple DOS 3.3 operating system. It covers those commands which permit the user to do tasks which call for the use of the disk drives.

Apple Computer, Inc. (1983). *Apple II ProDOS user's manual*. Cupertino, CA: Author.

An Apple reference manual detailing the Apple ProDOS operating system. It explains how to use the ProDOS User's Disk to manage information on disks.

Apple Computer, Inc. (1983). *Applesoft BASIC programmer's reference manual*. Reading, MA: Addison-Wesley.

A reference manual for the Applesoft BASIC programming language as used by the Apple II series of computers. It is for use by those that have some programming experience using the BASIC language.

Apple Computer, Inc. (1985). *The Applesoft tutorial*. Reading, MA: Addison-Wesley.

Apple Computer, Inc. (1986). *A touch of Applesoft BASIC*. Cupertino, CA: Author.

These manuals are for the beginning programmer who plans to use the Applesoft BASIC language to program the Apple II series of computers.

Blair, P. (1986). *Animation*. Tustin, CA: Foster Art Service Inc.

A very good how-to-do-it book on animation. It is well illustrated with Disney style characters.

Bowman, D.J. (1984). *The CAD/CAM Primer*. Indianapolis, IN: Howard W. Sams & Co., Inc.

An excellent source of information concerning CAD/CAM (more CAD than CAM) concepts, hardware, software, and terminology. It contains a very good glossary of CAD/CAM terms. This is a generic textbook not a specific CAD application textbook.

Apple Computer, Inc. (1987). *BASIC programming with ProDOS*. Reading, MA: Addison-Wesley.

Explains the ProDOS commands and shows how to incorporate them into Applesoft BASIC programs--a must for the BASIC programmer who is using the ProDos operating system.

Benton, R. & Balcer, M.S. (1988). *The official Print Shop handbook*. New York, NY: Bantam Books.

A book that provides ideas, tips, and designs for users of the *Print Shop* application software.

Berryman, G. (1979). *Notes on graphic design and visual communication*. Los Altos, CA: William Kaufmann, Inc.

Explains graphic design concepts with visual examples. An easy to read book covering designing with marks, symbols, logos, pictographs, and type. The stages of the graphic design process are also very well explained. An excellent resource book for many of the activities found in this curriculum guide, especially the one on desktop publishing.

Bertoline, G.R. (1985). *Fundamentals of CAD*. Albany, NY: Delmar Publishing, Co.

A comprehensive and generic CAD text. It has an excellent introduction to CAD systems in the first section. The second section describes a typical CAD system, and the third section contains instruction relative to creating a drawing using a CAD system-including appropriate drawing exercises. The last section contains a discussion of CAD applications. Each chapter has an excellent glossary. A good introductory text for grades 9 - 12.

Bitter, G., & Severe, S. (1986). *Programming in BASIC for the Apple II series*. Reading, MA: Addison-Wesley.

A very good BASIC text for beginners. It covers the basics without the clutter of those details associated with infrequently used commands.

Consumer Guide Editors. (1983). *Illustrated computer dictionary*. Skokie, IL: Publications International.

A good computer dictionary that is also well illustrated.

Fox, D., & Waite, M. (1984). *Computer animation primer*. New York: McGraw-Hill.

This book is in two parts. The first part is a good overview of computer animation for the more advanced student (the reading level and mathematics are advanced). The second part deals with animation on an Atari computer. Although the exercises are for the Atari, there is much useful information for someone interested in animation, but using a different computer.

Goetsch, D.L. (1983). *The CAD/CAM workbook*. Cincinnati, OH: South-Western Publishing Co.

Designed to be used with a traditional drafting text, this workbook can be used in a setting where there are no computers (according to the author). It exposes the student to the terminology and concepts of CAD/CAM. Each unit is designed as a self instructional module.

Goetsch, D.L. (1985). *Computer-aided drafting*. Englewood Cliffs, NJ: Prentice-Hall, Inc.

A follow-up text to *Introduction to Computer-Aided Design* by Goetsch. This text covers the subject better than Goetsch's introductory text. It has more breadth and depth and it leaves out some of the unnecessary information found in the earlier text. As such, it is a very good text for the CAD activity in this curriculum guide.

Hubbard, S. W. (1983). *The computer graphics glossary*. Phoenix, AZ: Oryx Press.

A comprehensive glossary on CAD/CAM. Contains terms unique to computer graphics as well as some of the more common CAD/CAM product names.

Jankel, A., & Morton, R. (1984). *Creative computer graphics*. Cambridge, England: Cambridge University Press.

An excellent publication presenting computer graphics from the pioneering efforts of the 50's to the current achievements of modern exponents from several countries. The book explores computer graphic images through the techniques and technology used to create them. It is well illustrated.

Jones, A.B. (1982). *I speak BASIC to my Apple*. Rochelle Park, NJ: Hayden Book Co.

A very good computer literacy book for the beginning computer student. It teaches concepts plus the BASIC language.

Kemnitzer, R.B. (1983). *Rendering with markers*. New York: Watson-Guptill Publications.

An excellent resource for students learning to illustrate with color. Although it explains how to illustrate with markers, the same principles can be applied to computer paint systems.

Lamit, L. G., & Paige, V. (1987). *Computer-aided design and drafting*. Columbus, OH: Merrill.

A well-written generic CADD textbook. It does a very good job of explaining the key concepts of a computer aided design system. Suitable for the more advanced student.

Laybourne, K. (1979). *The animation book*. New York: Crown.

A comprehensive text describing all types of animation and animation tools, techniques, and resources. The beginning chapters include a very good introduction to computer animation. The remaining portions of the book are also recommended for teachers and those students with an advanced interest in animation.

Martin, C.L. (1968). *Design graphics*. New York, NY: The Macmillan Company.

Although it has an old publication date, this is still an excellent reference for the teacher who is planning a presentation on illustration and rendering techniques in anticipation of students working with a paint or animation application program.

McNeil, D. (1987). *Compute!'s quick and easy guide to desktop publishing*. Greensboro, NC: Compute! Publications Inc.

An overview of the revolution known as desktop publishing. Includes a history of the field of printing, the terms used in printing, writing copy, designing the page, selecting type and graphics, and organization and dissemination of the final document. May not be suitable for students who do not read well. However, it is recommended as a reference for those using the desktop publishing activity.

Myers, R.E. (1984). *Microcomputer graphics for the Apple computer*. Reading, MA: Addison-Wesley.

An excellent source for the advanced Apple II graphics student. Contains explanations of most of the advanced techniques used by professional Apple graphics programmers. It can be purchased with a diskette containing those programs listed in the text.

McKissick, M. L. (1987). *Computer-aided drafting and design*. Englewood Cliffs, NJ: Prentice-Hall.

A generic CADD textbook for the more advanced student.

Minnesota School Districts Data Processing Joint Board. (1983). *Apple II microcomputer quick reference guide*. Minneapolis: The author.

A very handy fan-fold reference guide designed for students by teachers. For use with the Apple II, II+, and IIe. It covers some of the basic computer operation procedures.

Parker, R.C. (1988). *Looking good in print*. Chapel Hill, NC: Ventana Press.

An excellent generic reference for desktop publishing design. It emphasizes page design and can be used with any desktop application program. Includes sections on the elements of graphic design, illustrations of the elements through page makeovers, and a detailed discussion related to the different products of desktop publishing--newsletters, newspaper and magazine ads, brochures, business correspondence, etc.

The Public Domain Exchange. (1987). *The best of Apple public domain software, edition II*. Santa Clara, CA: the author.

A comprehensive listing of non-copyrighted public domain software, copyrighted public domain software, and shareware for the Apple II--software which is free or inexpensive. The software is catalogued by application. Of particular interest to teachers and students using this curriculum guide are the listings under art, CAD, and utilities.

Reid-Green, K. (1984). *Games computers play*. Bedford, MS: Digital Press.

A very good reference for the student who is a good programmer and wants to develop computer games with computer graphics. Computer code used in the book is structured and machine independent.

Sayre, C. L. (1986). *Design and graphics using the Apple computer*. Reston, VA: Reston Publishing Co.

A very good reference for those who want an in-depth understanding of graphics programming using Applesoft BASIC. Those choosing to use this book should be familiar with the BASIC language and the Applesoft graphic commands.

Schmieman, S. (1985). *MacPaint drawing, drafting, design*. Bowie, MD: Brady Communications Company, Inc.

This book may be of use for those teachers and students using some of the new Apple IIgs paint software that uses the *MacPaint* format. In addition to specific instructions in how to use the *MacPaint* functions, it has good ideas for painting activities and exercises.

Seymour, R.D., Ritz, J. M., & Cloghessy, F.A. (1987). *Exploring communications*. South Holland, IL: Goodheart-Willcox Co.

A very good resource for those studying the technological field of communications.

Sordillo, D. A. (1983). *The personal computer BASIC(S) reference manual.* Englewood Cliffs, NJ: Prentice-Hall.

A handy reference for those programmers who are interested in rewriting a computer program for a different personal computer. It covers and compares the different dialects of the BASIC language used by the IBM PC, Apple II, Atari, TRS Color, TRS Mod III, and the Commodore VIC-20 computers.

Spencer, D. D. (1986). *The illustrated computer dictionary (3rd ed.).* Columbus, OH: Merrill.

A comprehensive and well illustrated dictionary of computer terminology.

Ungar, J. (1985). *Rendering in mixed media.* New York, NY: Watson-Guptill Publications.

A good resource for the teacher who is preparing presentations on rendering for his or her students. Although the text deals with mixed media, many of the rendering principles apply to a computer paint system, which simulates many different media.

Voisinet, D.D. (1987). *Computer-aided drafting and design, concepts and applications.* New York, NY: Gregg Division of McGraw-Hill.

A generic and more advanced CAD textbook suitable for teachers and more advanced students. In addition to covering those topics found in Voisinet's introductory text, it covers some more advanced topics such as 3-D modeling, partial programming, the effects of color, and CAD/CAM.

Voisinet, D.D. (1986). *Introduction to computer-aided drafting (2nd edition).* New York, NY: Gregg Division of McGraw-Hill.

A generic introductory CAD textbook suitable for those 9 - 12 graders who are reading at their expected level. The text covers the topics CAD uses, equipment, and methods of use. It also covers methods of creating drawings, the typical CAD functions, and the CAD professional's tasks.

Williams, K., Kernaghan, B., & Kernaghan, L. (1983). *Apple II computer graphics.* Bowie, MD: Robert J. Brady Co.

A very good book for those who wish to do computer game type animation using Applesoft BASIC. The reader should be very familiar with Applesoft BASIC and the Apple operating system before trying to read. The reader should also understand hexadecimal and binary numbering systems--the first chapter deals with these topics.

Winkler, C. (1983). *The computer careers handbook.* New York: Arco Publishing Co.

This book provides an overview of the computer field, the jobs available, salaries paid, and the training and education needed. As with many computer career information books, it is slightly dated.

Periodicals

A+ Ziff Davis Publishing Company, One Park Avenue, New York, NY 10016. Subscriptions should be sent to *A+*, P.O. Box 52324, Boulder Colorado 80321-2324. A monthly publication.

Information, tutorials, product reviews, and industry news about the Apple II product line. Suitable for grade levels, 9 - 12.

Cadence. Cadence Publications, 12611 Research Blvd., Suite 367, Austin, TX 78759-9990. A monthly publication.

A magazine for the *AutoCad* user. Contains articles on *AutoCad* applications and helpful techniques, and information about compatible hardware and software. Suitable reading for teachers and students who are very familiar with *AutoCad* software.

Classroom Computer Learning. Peter Li, Inc., 2541 E. River Road, Dayton, Ohio 45439-9988. Published monthly except June, July, August, and December.

Articles, product news, shoppers guide, and answers to commonly asked questions about computers and their use in the elementary and secondary classroom. An excellent publication for teachers who use computers in their classrooms.

Computer Graphics Today. Media Horizons Inc., 50 West 23rd Street, New York, NY 10010. Subscriptions should be sent to *Computer Graphics Today*, POB 656, Holmes, PA 19043. A monthly publication.

The official publication of the National Computer Graphics Association. Subscription is free to members of the association. Contains announcements about computer graphics products, publications, and events. Suitable reading for teachers and most high school students who have a high interest in the field.

Computer Graphics World. PennWell Publishing Company, P.O. Box 1112, 119 Russell Street, Littleton, MA 01460. Subscriptions should be sent to *Computer Graphics World*, P.O. Box 112, Tulsa, OK 74101. A monthly publication.

Industry news, product announcements, and product reviews about computer graphics applications. A good general interest publication designed for the user. Suitable for teachers and junior and senior level students.

Computer Pictures Magazine. Electronic Pictures Corporation, 2 Village Square West, Clifton, NJ 07011. Published bimonthly.

A technical magazine with news and trends in the computer graphics field. Emphasis is placed on computer visual art and design, and animation for film and video. Suitable reading for teachers and advanced students.

IEEE Computer Graphics and Applications. A publication of the IEEE Computer Society Executive Committee, The Institute of Electrical and Electronics Engineers, Inc., 1730 Massachusetts Ave., NW, Washington, DC 20036-0101. Subscriptions should be sent to IEEE Computer Society, 10662 Los Vaqueros Circle, Los Alamitos, Ca 90720. A monthly publication.

A very technical publication usually containing articles on computer graphics algorithms and mathematical techniques. It also contains industry news and new product announcements. May be of interest to those senior high school students with an advanced mathematical background.

InCider. CW Communications Peterborough, Inc., Route 101 and Elm Street, Peterborough, NH 03458. Subscriptions should be sent to Subscription Department, P.O. Box 911, Farmington, NY 11737.

An excellent periodical for the Apple II computer family user. It is highly recommended for both students and teachers. It does a very good job of rating software. It also has well written articles that are timely and helpful both for the programmer and the application software user.

Publish. PCW Communications Inc., 501 Second Street, San Francisco, CA 94107. Subscription Information available from Subscriber Services, P.O. Box 55400, Boulder, CO 80322. A monthly publication.

A desktop publishing periodical with a hands-on approach. Includes feature articles dealing with user's tips, applications, and profiles. It also includes reviews of desktop publishing software. Suitable for anyone interested in desktop publishing.

Software

Barbatano, M., Darooge, M., McCullough, E. & Walker, D. (1986). *Take 1 and Take 1 programmer's toolkit*. Available from Baudville, 1001 Medical Park Drive S.E., Grand Rapids, MI 49506 or a local computer software dealer.

A top rated standard hi-res animation application program. It has outstanding features and is easy to use. Allows the user to create "actors", backgrounds, and sounds to create scenes. Scenes can be combined and made into movies with fade-ins, fade-outs, and sound effects. It is reasonably priced and will run on an Apple II plus, IIe, IIc, or IIgs with 64K.

Darooge, M., McCullough, E. & Walker, D. (1984,1988). *Blazing paddles*. Available from Baudville, 1001 Medical Park Drive S.E., Grand Rapids, MI 49506 or a local computer software dealer.

A top rated standard hi-res paint application program for an Apple II plus, IIe, IIc, or IIgs with 64K of RAM. It is reasonably priced, works with a variety of input devices, and is very easy to use.

Langer, V. (1985). *MATC-CAD* Available from the Milwaukee Area Technical College, 1015 North Sixth Street, Milwaukee, Wisconsin 53203.

An excellent application software package for CAD education. The software was developed under a grant from the National Science Foundation specifically for training purposes. It simulates Computervision's *CADDs* software which is very popular in industry. Comes with excellent teaching materials. There are 64K and 128K Apple versions and an IBM PC version.

Lissner, R.J. & Apple Computer, Inc. (1983-86). *AppleWorks, V.2.0*. Available from the Claris Corporation, 20525 Mariani Avenue, Cupertino, CA 95014 or a local computer software dealer.

An integrated application program containing a word processor, a data base manager, and a spreadsheet. Teachers may find the word processor very useful with several of the activities in this curriculum guide, especially the desktop publishing activity. Students may also find the data base manager or the word processor useful in keeping a record of their computer graphic terms. It is also an excellent tool for classroom management. It can be used for laboratory inventory, test files and test making, and as a computerized gradebook. Requires an Apple IIe, IIc, or IIgs with 128K of RAM.

Milliken Publishing Co. (1988). *Medley*. Available from the Milliken Publishing Co., 1100 Research Blvd., St. Louis, MO 63132-0579 or a local computer software dealer.

A desktop publishing application software package that uses Macintosh-like menus and windows. It includes word processing with an 80,000 word thesaurus and spell checker, graphics, and page layout programs all-in-one. Comes with clip art and a back-up disk. Requires an Apple IIgs with 512K of RAM memory.

Mullen, Fred L. (1983). *Autograph*. Available from the Public Domain Exchange, 2074C Walsh Avenue, Dept 20, Santa Clara, CA 95050. Order disks 310,311, & 312.

Inexpensive CAD software. This is public domain software that can be obtained from the Public Domain Exchange for use on the Apple IIe. It is not very user-friendly.

Ohmert, S. (1986). *Animate*. Available from Broderbund Software, Inc., P.O. Box 12947, San Rafael, CA 94913-2947 or a local computer software dealer.

A top rated double hi-res animation application program. The program has the necessary painting tools to create and edit animated cells and background scenes. Scenes can also be drawn with other software such as *Dazzle Draw* and then used with *Animate*. It uses pull down menus and windows similar to those found in Macintosh applications. The software comes with good documentation and a well illustrated book on animation. It will run on an Apple IIe, IIc, or IIgs with 128K.

Scholastic, Inc. (1987). *Bank Street writer III*. Available from Scholastic, Inc., 730 Broadway, New York, NY 10003 or a local computer software dealer.

Considered to be the best word processor for home and school use, this software is extremely easy to learn and use. It contains a spell checker and a thesaurus. Requires any Apple II computer with 128K of RAM.

Silva, D. (1988). *Deluxepaint II*. Available from Electronic Arts, 1820 Gateway Drive, San Mateo, CA 94404 or a local computer software dealer.

A top rated super hi-res graphics paint application program. This program has all of the functions normally found in a paint program plus some not expected in an Apple application. It will run on an Apple IIgs with 768K of RAM.

Snider, D. (1984,1987). *Dazzle draw*. Available from Broderbund Software, Inc., P.O. Box 12947, San Rafael, CA 94913-2947 or a local computer software dealer.

A top rated double hi-res paint application program. This is an excellent program with which to introduce the user to the capabilities of a computer graphics 2D paint application. Contains most of the functions found on a much more expensive system. Will run on an Apple IIe, IIc, or IIgs with 128K.

Softsync, Inc. (1987). *Personal newsletter*. Available from Softsync, Inc., 162 Madison Ave., New York, NY 10016 or a local computer software dealer.

This desktop publishing software will allow the user to create many kinds of documents, but it is especially suited to creating newsletters. The program has graphics functions for drawing borders and fill patterns, it will permit from one to ten columns for text layout, has six fonts in ten type styles, and comes with clip art. You may also import graphics from *Print Shop, The Newsroom, Dazzle Draw*, or a Thunderscan (digitizing) unit. It includes the drawing tools to draw your own or edit imported drawings. Requires an Apple IIe (with ROM enhancement), IIc, or IIgs with 128K and an 80 column board.

Springboard Software. (1988). *Springboard publisher*. Available from Springboard Software, 7808 Creekridge Circle, Minneapolis, MN 55435 or a local computer software dealer.

A desktop publishing application that emulates the well known *Pagemaker* application on the Macintosh. This software has three working modes--page processing, word processing, and graphics. It was designed with school and home applications in mind. The clip art that comes with it called, "Works of Art", contains graphics related to educational and holiday themes. Additional clip art may be purchased. Files may be imported from *Mouse Paint, Springboard Art*, and ASCII text files.

Springboard Software. (1985). *The newsroom*. Available from Springboard Software, 7808 Creekridge Circle, Minneapolis, MN 55435 or a local computer software dealer.

This is a software classic. It contains word processing, graphics, banner, and telecommunications functions. Although limited when compared to some of the newer page and newsletter processing software, it is still an effective educational tool. A lot of clip art is available for use with it and it comes with comprehensive documentation including suggestions for gathering information, writing stories, and organizing a classroom newspaper. Requires an Apple II Plus, IIe, or IIc with 64K and one disk drive (two are suggested). Works with any number of matrix printers. Also available for the IBM PC.

StyleWare. (1987). *Multiscribe V.3.3*. or *Multiscribe GS V.3.0*. Available from StyleWare, 5250 Gulfton, Suite 2E, Houston, TX 77081 or a local computer software dealer.

A full featured word processing program that uses double hi-res graphics and proportional spacing to emulate the *MacWrite* word processor used on the Macintosh computer. It uses the pull down menus with a mouse or the keyboard and has a multitude of fancy fonts. The GS version is especially nice with its spell checker and thesaurus, graphics capabilities, and the ability to directly import *AppleWorks* files. *Multiscribe 3.3* requires an Apple IIe, IIc, or IIgs with 128K and an 80 column card. *Multiscribe GS 3.0* requires a IIgs with 512K.

Timeworks. (1988). *Publish it!* Available from Timeworks, 444 Lake Cook Road, Deerfield, IL 60015 or a local computer software dealer.

This desktop publishing software has a Macintosh type interface with pull-down windows, icons, etc. and a full featured word processor. It will allow the user to import files from *Dazzle Draw, Mouse Paint, Beagle Graphics, AppleWorks, Bank Street Writer*, and ASCII files from other word processors. It can be used with a mouse or a joystick.

Versacad Corporation. (1987). *Entry level CADApple*. Available from Versacad Corporation, 7372 Prince Drive, Huntington Beach, CA 92647 or a local computer software dealer.

An entry level CAD software application package that uses a subset of the functions available in the popular *CADApple* software. Comes with a textbook and an audio tape.

Version Soft, Inc. (1986). *Paintworks plus*. Available from Activision, Inc., P.O. Box 7287, Mountain View, CA 94039 or a local Apple computer dealer.

A paint application program and an animation application program all in one. The paint program has the same painting tools as MacPaint and it uses the pull down menus and windows found in Macintosh programs. It has the ability to use 16 colors in one painting selected from a palette of 4096 colors. Easy to use and comes with good documentation. Can be used on an Apple IIgs with 512K.

Videotapes

CIM Solutions. (1986). *Computer aided design: a video evaluative report I*. Available from CIM Solutions, University Station, P.O. Box 7041, Provo Utah 84602.

CIM Solutions. (1986). *Computer aided design: a video evaluative report II*. Available from CIM Solutions, University Station, P.O. Box 7041, Provo Utah 84602.

These two videotapes evaluate a variety of popular computer aided design systems by having the viewer look over the shoulder of a CAD professional as he/she draws a benchmark object using each of the different systems. A good idea; however, the professionals are not always very well trained on the system they are using and some of them make some serious drafting errors. The tapes do allow the viewer to see the different functions supplied by systems.

DCA Educational Products, Inc. (1982). *Apple IIe, VHS*. from the *Computer operations series*. Bedminster, Pennsylvania 18910: The author.

An older video tape that does an excellent job of introducing a new user to the Apple family of computers. It demostrates the basic tasks of turning the computer on, using the disk drive, and running programs. It may not be as appropriate for students who are using the Apple IIgs as it is for those using the Apple IIe or IIc.

Dupont, C. & Cinemagic Productions. (1987). *Computer magic*. Available from Cinemagic Productions, 537 Jones Street, #898 San Francisco, CA. 94102.

An excellent presentation by one of the pioneers of computer animation, Mr. Robert Abel. He takes the viewer behind the scenes to describe the production of some of the more popular computer animated T.V. commercials.

Electronic Pictures Corporation. (1987). *Computer graphic special on TV commercials and animation*. Available from Frost and Sullivan, Inc., Department R-1, 106 Fulton Street, New York, NY 10038.

An exciting anthology of some of the best computer animation ever produced, presented with fascinating explanations of many of the creative processes and production techniques actually used.

Electronic Pictures Corporation. (1983). *Creative computer graphics*. Available from Frost and Sullivan, Inc., Department R-1, 106 Fulton Street, New York, NY 10038.

An excellent introduction to computer graphics. The video surveys the different applications of computer graphics, defines some key terminology, and explains some of the key technological advances in the field.

Frost and Sullivan, Inc. (1986). *SIGGRAPH '86*. A Frost and Sullivan video report. Available from Frost and Sullivan, Department R-1, 106 Fulton Street, New York, NY 10038.

A guided video tour of the SIGGRAPH '86 Exhibition. SIGGRAPH is the Special Interest Group on Graphics of the Association of Computing Machinery (ACM). The tour features brief presentations of more than 40 computer graphics products and in-depth presentations of about 20 more products.

Frost and Sullivan, Inc. (1987). *Workstations/video text report (VTR) 001*. A Frost and Sullivan video text report. Available from Frost and Sullivan, Department R-1, 106 Fulton Street, New York, NY 10038.

A video on state-of-the-art computer graphics hardware in the workstation category. Also contains a verbatim transcript of the narration and commentary. This is a very technical video and should only be shown to students after a discussion of some key computer graphics hardware terminology. With the presentation of each type of workstation there are examples of the output that can be created on the workstation. This makes it a very interesting video for anyone interested in computer graphics.

Frost and Sullivan, Inc. (1987). *Graphic arts systems/video text report (VTR) 002*. A Frost and Sullivan video text report. Available from Frost and Sullivan, Department R-1, 106 Fulton Street, New York, NY 10038.

A video on state-of-the-art computer graphics 2D paint and 3D modeling systems. Also contains a verbatim transcript of the narration and commentary. This is a very technical video and should only be shown to students after a discussion of some key computer graphics terminology related to paint and modeling systems. With the presentation of each type of system, there are examples of the output that can be created on the system. This makes it a very good video for anyone interested in computer graphics.

Herr, L., & Zaritsky, R. (1988). *Visualization: state of the art*. A special issue of the ACM/SIGGRAPH Video Review--Volume 30. Association for Computing Machinery, 11 West 42nd Street, New York, NY 10031.

A comprehensive coverage of state-of-the-art computer graphics. This video is over two hours in length and covers historical events (Sketchpad at MIT, 1964), new developments and trends in hardware, developments in graphics programming and rendering, computer graphics systems, a special look at computer graphics in scientific computing and medical imaging, and a look at the future of computer graphics. The video features most of the best known computer graphic scientists as speakers and plenty of computer generated graphics. An excellent video; it should be viewed by all computer graphics enthusiasts.

Note: The teacher should allow plenty of time to review the video. There are some segments that are only appropriate for the most advanced students. Therefore, he or she should carefully select those segments that are appropriate for his or her students. The video comes with a well documented script and topical outline which will help in the review.

Massachusetts Institute of Technology. (1987). *MIT media laboratory*. Cambridge, MA: Available from the author.

A videotape with segments on each of the major research projects at MIT's media laboratory where some of the most exciting computer application research is taking place. Much of the research utilizes computer graphics.

National Computer Graphics Association. (1987). *1987 International computer animation competition winners*. Available from NCGA, 2722 Merrilee Drive, Fairfax, VA 22031.

National Computer Graphics Association. (1988). *1988 International computer animation competition winners*. Available from NCGA, 2722 Merrilee Drive, Fairfax, VA 22031.

Videotapes of most of the 1987 and 1988 International Computer Animation Competition winners. This competition is sponsored each year by the National Computer Graphics Association and each year the NCGA releases a videotape of most of the animation sequences for which awards were presented. Some winners are not included because they were originally developed for customers who do not wish to release them.

Ocean Communications, Inc. (1987,1988). *On becoming a desktop publisher*. Ocean Communications, Inc., 1641 N. First St., Suite 160, San Jose, CA 95112.

Includes two detailed videotapes. One is a tutorial including a discussion of computers, layout, artwork, printers, fonts, large screen displays, and scanners used in desktop publishing. The second is a demonstration tape where the better known page layout and artwork software packages and the three Apple laser printers are presented. Even though these tapes are prepared for business use of desktop publishing with the Macintosh and IBM PC compatibles, they can give the student a very good overview of the field.

SIGGRAPH/ACM. *SIGGRAPH video review--issues 1 through 24*. Association for Computing Machinery, 11 West 42nd Street, New York, NY 10031.

These issues illustrate the latest concepts in computer graphics and interactive techniques. Begun in 1980 with issue 1, they have been made each year since. Available on 3/4" U-matic and VHS.

Silicon Graphics Computer Systems. (1987). *Superworkstation demonstration*. Silicon Graphics, Corporate Headquarters, 2011 Suerlin Road, Mountain View, CA 94043.

A promotional video on Silicon Graphics Workstations. It surveys very well the current applications related to CAD and CAE. An excellent video for students interested in state-of-the-art mechanical engineering applications.

GLOSSARY

absolute coordinates.

The value of the X,Y, and Z Cartesian coordinates with reference to the origin of the coordinate system.

AEC.

An abbreviation for Architectural, Engineering, and Construction. A segment of industry that uses a special type of computer aided design software for architectural design, facilities planning and management, highway design, and many other architectural or construction related applications.

algorithm.

A step by step procedure for solving a problem. Often used in computer programming to describe a procedure with a finite number of well defined steps for solving a problem within a program.

aliasing.

The stair stepping effect that is caused by drawing a non-horizontal or non-vertical line on a raster refresh display device or any other raster type of device. The poorer the resolution of the device the more pronounced the aliasing. Sometimes referred to as the "jaggies."

alphanumeric.

Refers to all of the characters in a standard alphabet, numerals 0-9, the control characters, mathematical symbols, punctuation marks, and other characters and symbols found on a conventional computer keyboard.

anti-aliasing.

A process where the visual effects of aliasing are reduced.

application software.

A set of computer programs design to solve problems or meet the requirements of a specific job or industry. In computer graphics, such software may be designed specifically for designing machine devices (CAD), document processing (Desktop Publishing), rendering a 2D image (Paint System), architectural drawing (AEC), etc.

array.

A series of terms, values, or elements arranged in a meaningful way. In a computer program, it refers to an area or file where data is stored. Also referred to as a matrix.

aspect ratio.

The relationship between height and width. In computer graphics it can be used to describe the viewport of a display device, the window of the graphic data base, text, or any graphic element. RATIO = HEIGHT/WIDTH or RATIO = WIDTH/HEIGHT.

assembly language.

A low level computer language. Consists of mnemonic statements that have a one to one correspondence with specific machine language instructions of the computer.

B-spline.

Refers to a simulated curve or curved surface generated by a graphics program. The generated curve or curved surface consists of a large number of minute vectors calculated to pass through a number of given points.

BASIC.

An acronym which stands for Beginner's All-purpose Symbolic Instruction Code. Originally developed at Dartmouth College, it is one of the more commonly used high level programming languages for personal computers.

batch processing.

Refers to a method of processing computer jobs, wherein the jobs are processed consecutively and without human interference. The opposite of interactive.

binary.

A characteristic where only two alternatives exist. Digital computers use a binary numbering system. This is a numbering system where the base is 2, and there are only two counting numbers, a 0 and a 1.

binding.

The act or method used to fasten pages of a document together usually with a document cover. Common methods of binding are saddle stitching, spiral binding, perfect binding, and edition binding.

bit map.

That portion of the computer's memory devoted to describing the image being displayed on the display device. With a raster refresh display device, there is a one to one correspondence between a bit or a group of bits in the frame buffer or bit map and a pixel on the display device screen.

bit.

A contraction of "binary digit." A single character in a binary number, either a 0 or a 1. The smallest unit of information in a digital computer.

byte.

A group of binary digits or bits. Sometimes referred to as a word. Usually consists of eight bits.

CAD.

An acronym for Computer Aided Design. A computer system application that is used for designing and drafting products.

CAD/CAM.

An acronym that stands for computer aided design/computer aided manufacture. Refers to an integrated computer system that is capable of controlling the design and the manufacture of a product from one data base.

CADD.

An acronym for Computer Aided Design and Drafting. Some prefer this acronym to CAD when describing a computer system application that is used for designing and drafting products.

Cartesian coordinates.

A system of defining and locating points in 2D and 3D space named for the mathematician, Rene Descartes. Using this system, a point is defined by its location from the origin relative to a set of mutually perpendicular axes, X, Y, and Z. This system is used in all computer graphics applications for locating graphic elements.

CIM.

An acronym that stands for Computer Integrated Manufacture. A computer system where not only is the manufacture and design of a product under computer control (CAD/CAM); but also inventory control, purchasing, inventory and production control, quality assurance, etc.

clipping.

That function performed by a computer graphics program when it discards graphic data that fall outside a specified boundary (the window). In some cases, this may necessitate that new endpoints be computed for the boundary intersections. If this task were not performed, wraparound could occur on the display device screen or an error message may halt execution. Wraparound is when lines are drawn on the display device screen in an uncontrolled manner.

compiler.

Computer software that translates a high level language such as FORTRAN or BASIC into machine language that the computer understands. After the program has been compiled, the machine language compilation, referred to as object code, is executed.

computer animation.

The use of a computer to generate video, film, or computer displayed real time sequences. Animation is the presentation of many still images in rapid succession so as to create the illusion of smooth fluid motion.

computer graphics system.

Refers to all of the software, hardware, and documentation necessary for a computer graphics application.

computer graphics.

That branch of computer technology that concentrates on the creation of graphic images for many different applications to include computer-aided design, desktop publishing, presentation graphics, computer-aided animation, medical graphics, and image processing.

CPU.

An abbreviation for Central Processing Unit. That portion of a computer that contains all of the hardware that controls and executes the programming instructions--the brains of a computer. In a microcomputer, the CPU may be on a single chip called the microprocessor.

CRT.

An abbreviation for Cathode Ray Tube. A vacuum tube similar to the one used in a T.V. It is the technology used in most computer graphics display devices. An electron beam draws images by focusing on a portion of a screen coated with phosphor. The excited phosphors glow for a short period of time, then they must be recharged (refreshed). CRTs used in raster refresh or vector refresh devices are refreshed at least 30 times per second.

cursor.

A position indicator used to show where data will be entered on a display device. Usually represented by crosshairs or a block of blinking light.

diffused light source.

When rendering an object, the term used to describe a light source that is widespread. Multiple light sources or a light source that has been spread by the act of passing through a diffuser--the light source on a cloudy day.

digitizer.

The computer graphics hardware that is used to convert analog data into digital data--usually a graphics tablet. See digitizing.

digitizing.

The act of converting analog data into digital data. This is usually done with a graphics tablet. A drawing is taped to the tablet surface and the graphic image is traced with a puck or stylus. As the image is being traced, digital data is fed into the computer. 3D digitizing is also performed in some applications.

display buffer.

Also known as a frame buffer. That portion of the computer's RAM memory in a raster refresh computer graphics system that contains data needed to refresh the screen. A raster refresh display device's screen must be refreshed (redrawn) at least 30 times per second or the screen's image will fade and flicker will occur. Without a more permanent record of the screen's contents, an image could not be maintained on the screen long enough for the user to view it. See bit map.

display device.

That computer graphics hardware that produces the graphic image. The most common display devices are pen plotters, matrix printers, electrostatic plotters, and cathode ray tubes (CRTs); however, the term is used most frequently to refer to the CRT.

dot matrix printer.

An output device that is commonly used with personal computers. It is capable of creating text and graphics equally well. The image or text is created when the head (composed of a matrix of electronically controlled pins) strikes the ribbon and paper. Generally, the greater the number of pins in the matrix, the higher the quality of the printed image.

drum plotter.

A type of plotter (output device) that produces drawings on media that are fixed to a rotating cylinder. It can refer to any type of plotter that uses a rotating cylinder, but usually is used to refer to a pen plotter which uses a rotating cylinder.

electronic air brush.

That function in paint application software that simulates an air brush or spray can. An air brush sprays ink on the media for a unique effect in illustrating or painting.

electrostatic plotter.

A computer graphics output (display) device that generates an image by placing an electrostatic charge on the medium, in a raster format, and then passing the medium through a bath of toner particles. An operation very similar to that used by most plain paper copiers.

endpoint.

The extreme limits of a line, usually defined by Cartesian coordinates. The logical conclusion of an operation.

faceted polygons.

The plane surfaces that are used in computer graphics to render a solid figure. Rendering a figure in this manner results in an object that looks like a faceted gem.

fatal error.

A computer programming error that causes the program to quit executing.

file.

A collection of information (data) stored with a designated name on a mass storage device such as a disk.

flatbed pen plotter.

A computer graphics electromechanical output device where the media is in a stationary position on a flat surface and the image is drawn with a pen or pens held by a carriage traveling parallel to the surface.

flicker.

A condition that occurs on a CRT refresh display device when the refresh rate is slowed to the point where the human eye can detect the screen flashing. There is a noticeable decay in the luminance of the phosphors. This can happen on any refresh device; although, usually for different reasons.

flip book.

A book with an assembly of still images. By rapidly flipping through the book, an animated sequence can be viewed. A Kinetograph.

floppy diskette.

A magnetic mass storage device consisting of a flexible disk encased in a plastic or cardboard envelope. Used for random-access storage.

format a disk.

The task of having a disk prepared to receive data from a particular computing system. Each computer system formats (arranges) data on a disk in a unique manner. Therefore, a disk must be formatted on the computer system with which it is to be used.

FORTRAN.

A acronym that stands for FORmula TRANslation. A high level computer programming language that is very popular especially in computer graphics and engineering applications.

frame buffer.

A special portion of RAM memory designed to store a digital description of a computer image. Most usually used to describe the image being displayed on a raster refresh display device screen and from which the screen is refreshed.

function.

A process in an application program, such as drawing a line, that is executed when a specific command is given, a key is pressed, or a menu button is selected.

graphic transformation.

A modification made to a graphic image or its data base. The basic transformations are scaling, rotation, translation, reflection, perspective, and shearing.

graphics tablet.

A computer graphics input device consisting of a flat tablet and a stylus or puck. It converts graphical data (usually a drawing taped to the tablet surface) into binary data for the computer or it is used to control the screen cursor.

greeking.

The description given to illegible scrawl that results when an actual size desktop publishing page is changed to full page layout. The prose is reduced to a point that it is no longer legible.

grid.

A very common type of computer graphics input constraint. Input data is transformed (rounded off) to correspond to a grid intersection. The grid, any uniform pattern of dots or intersecting horizontal and vertical lines, can be displayed on the screen to aid the user. However, it is not necessary that the grid be displayed to have the input data transformed.

hard disk.

A fast computer mass storage device that can be housed in a separate case or within a computer. A single hard disk has a storage capacity of several million characters or bytes of information. This device utilizes a disk that is hard or firm as compared to a floppy disk.

hardware.

Physical computer equipment--electronic, magnetic, and mechanical components of a computer system. Contrasts with computer firmware, software, and courseware.

hexadecimal.

A numbering system with a radix of 16. The counting numbers in this system are 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E, and F. The hexadecimal number of 3B equals the decimal number of 59. $3 \times 16_{16} + B$ (or 11_{10}) $\times 1 = 59_{10}$.

high level language.

Any programming language such as BASIC, FORTRAN, PASCAL, and C that allows statements to be written in a familiar notation (English-like) code rather than machine or assembly code. High level language code must be translated to machine code before the computer can execute it. This is most usually done by a software interpreter or a compiler.

high resolution graphics display buffer.

See frame buffer.

I/O device.

I/O is an abbreviation for Input/Output. I/O devices are those hardware devices that are used with a computer system to allow data to be passed to or received from the central processing unit. Input devices used in computer graphics include light pens, tablets, mice, thumbwheels, track balls, touch screens, and joysticks. Graphic output devices include plotters, film recorders, matrix printers, laser printers, and various mass storage devices.

IGES.

An acronym that stands for Initial Graphics Exchange Specification. This is a standard adopted by many CAD vendors. The standard specifies how drawing data is to be formatted when the data is exchanged among the different CAD systems.

image enhancement.

The accentuation of a graphics image or part of an image by highlighting, zooming, blinking, or coloring.

input.

See I/O device.

interactive.

When used to describe a computer system, it is any system that yields an immediate response to the user's input. Where the computer and the user are in constant communication. Contrasts with batch processing system.

interpreter.

Software that acts as a language translator. It converts source language written in BASIC or another high level language into machine language code and it is executed immediately. Unlike a compiler, each statement is translated and executed before the next statement is processed.

isometric drawing.

A pictorial view of an object in which three faces of the object can be seen. This view is achieved by projection when the object is positioned so that its faces all make an angle of 35 degrees, 16 minutes to the projection plane. The resulting image's major axes are equally spaced (120 degrees apart) and equally foreshortened. However, when drawn by hand or a 2D computer graphics system, the axes are not usually foreshortened.

jaggies.

The stairstepping or saw-toothed effect achieved when drawing diagonal lines or simulated curved lines on a raster type device--a raster refresh display device, a matrix printer, or an electrostatic printer. The formal term describing this effect is aliasing.

joystick.

A computer graphics input device consisting of an upright stick and a button. Movement of the stick forward, backward, right, left, or in some combination manipulates the cursor on the screen. Very popular with computer or video games.

justification.

The act of arranging lines of digits or text to fit a prescribed pattern. Left justification results in groups of lines being flush on the left but with a jagged right edge. A justified group of lines has both right and left edges flush. Lines can also be center justified and right justified.

kerning.

A process used in printing where a piece of type with a certain character such as an L or a T is constructed in a manner to allow it to overlap the next piece of type (ligature). This results in printed copy with equal white space between characters, a result that cannot be achieved by just using proportional spacing.

kilobyte.

A measure of computing memory space. Specifically, 2 raised to the tenth power, or 1024 bytes. Kilo usually is thought to be 1000; however, in computing 1024 is the closest practical value. Therefore, 64K bytes is 64 x 1024 or 65,536 bytes of information. Abbreviated as kb. Compare to megabyte, gigabyte, and tenabyte.

layers.

A function within a CAD system that allows the user to segregate entities and then control them individually. They can be displayed or plotted together or separately. Layers can be thought of as separate sheets of tracing paper that are overlayed on a basic drawing. A technique used in architectural drawing when the basic floor plan is placed on one layer, the electrical plan on another, the plumbing plan on a third, etc.

leading.

Pronounced LED-ing. In printing and desktop publishing, refers to the spacing between lines of text. When all type was set by hand, spacing was achieved with thin strips of lead, thus the name.

letterspacing.

This is a technique used in printing where individual letters are slightly separated. It is the opposite of kerning.

light pen.

A graphics input device that is used in conjunction with the display screen. Shaped like a pen, it contains a photoelectric cell that detects light being emitted from the screen. With the proper software, it can be used to detect objects on the screen or draw images on the screen.

LOAD.

An Apple DOS and ProDOS command used to read a program from a file on a disk into the computer memory. Usually used in immediate mode. The key word LOAD is followed by the name of the filename under which the program is stored.

machine language.

The language understood by computers. Programs written in this language require no further interpretation in order for the machine to execute. Machine language instructions and data are presented in binary code.

mainframe.

A classification of computer central processing hardware representing computers that are very large and expensive. That classification between minicomputers and supercomputers.

mass storage device.

An output device that is capable of storing large amounts of data magnetically. It stores data in a non-volatile state. A term used to describe devices such as floppy disk drives, tape drives, and hard disk drives.

matrix printer.

A printer that is capable of printing text and graphics. It produces the image by using a printing head that is composed of a matrix of small pens capable of making dots when the head impacts the ribbon and paper. The pens can be selected in the proper arrangement to display any number, symbol, or alphabetic character.

matrix.

A two dimensional array of data. A matrix is a very convenient way to store data representing the coordinates of a graphics image.

megabyte.

A measure of computing memory space. Specifically, 2 raised to the 20th power or 1,048,576 bytes; 1024 kilobytes. See kilobyte elsewhere in this glossary.

menu.

A list of alternative processing functions from which the user can choose. In a computer graphics application, these functions can be listed as words or icons on the display device screen or they can be placed on a graphics tablet.

microcomputer.

A classification of computers. Usually the smallest and least expensive classification. It uses a microprocessor as a central processing unit.

microprocessor.

The central processing unit for a microcomputer. An integrated chip that contains arithmetic logic, control logic, and memory control elements necessary in a central processing unit.

minicomputer.

A classification of computer hardware. Refers to those computers that are smaller and less expensive than a mainframe, but larger and more expensive than workstations and microcomputers.

mirror.

Another term used to describe the basic transformation, reflection. A display of a graphic image in exactly the reverse orientation from its original position.

modeling.

The construction of a 3D object from 3D primitives (cylinders, cones, pyramids, prisms, etc.) by performing basic transformations such as scaling, translation, reflection, and rotation. Refer to wire frame, surface, and solid modeling.

mouse.

A graphic input device. Most commonly, a hand size housing containing a ball underneath and buttons on the top. The screen cursor is tracked by the movement of the device on a relatively small surface.

National Computer Graphics Association.

A professional association of computer graphics professionals from business, industry, and education. The association has headquarters in Arlington, Virginia and sponsors local chapters in many states.

object blend sequence.

The use of a computer to generate intermediate frames given key pictures or frames representing the extremities of a movement in an animation sequence.

object code.

The machine language computer code that is output from a compiler or assembler. It can be executed by the computer or is ready for further processing to produce executable code.

offset printing.

A method of printing that is very common. It is based on the principle that oil and water do not mix.

operating system.

That computer software that controls the basic operation of the computer. This software (sometimes referred to as OS) controls input and output (DOS), the execution of programs, debugging, data management, etc.

optical character readers.

A computer input system that accepts printed documents and identifies the characters by their shape. Some systems can only recognize characters when certain fonts have been used while others can even recognize handwriting.

origin.

The point where the X and Y axes of a Cartesian coordinate system intersect (X=0.0, Y=0.0, & if 3D Z=0.0). Usually the bottom left hand corner of the screen. However, Apple has chosen to use the upper left hand corner as the origin.

orthographic drawing.

A technical drawing that is derived by a form of projection where the projectors are parallel and make an angle of 90 degrees to the plane of projection. Types of drawings derived in this way are multiviews (2 and 3 views, etc.) and axonometrics (isometrics,dimetrics, and trimetrics).

output.

Data that is transferred to a storage device or some output device (plotter, printer, etc.) after being processed by the computer. Also, the final form of data that has been processed by a computer (a plot, a printout, an animated sequence).

Pascal.

A high level computer language named after the French mathematician, Blaise Pascal, who built the first desk-type-calculator adding machine in 1642. Pascal, the computing language, is an easy to use language that is highly structured and powerful. It is very popular in computer science college programs.

passive graphics.

A computer graphics program or application that transpires without intervention from the user. The opposite of interactive graphics.

pattern recognition.

The recognition of patterns, forms, or configurations by a computer system. The visual system used on a robot is an example of the practical use of such a system.

persistence of vision.

A physiological phenomenon which comes into play when humans watch T.V. or go to the movies. When pictures move faster than 18 to 24 times per second the images appear to blend together. This occurs because a single image flashed at the eye is retained longer by the brain than it is registered on the retina. The brain is still registering the last image when a new image is flashed; thus, the two images are blended together.

perspective drawing.

A type of technical or artistic drawing that most closely resembles how humans see images. This type of drawing is achieved by projecting an image on the plane of projection by projectors that are not parallel, but converge at some point designated as the station point (location of the observer). A type of drawing that is difficult to achieve on a drawing board, but relatively easy for a computer to execute.

photocopying.

The most popular method of document copying. Photocopiers can be found in most modern offices and are often referred to as plain paper copiers because special papers are not needed to achieve satisfactory copies. A copying process introduced by Xerox.

pixel.

The smallest element on a raster refresh type of computer display device that can be controlled by the computer. A raster refresh display device consists of a matrix (multiple rows and columns) of these elements. Individually, they can be turned off, turned on, or the color can be changed in order to draw different images. Short for "picture element."

point of intersection.

The position in space where a line crosses another line or the surface of a plane or solid.

polyline.

A graphic primitive or entity that consists of multiple lines joined end to end, but not closed so as to constitute a polygon.

program.

A series of instructions written for the computer which causes it to process data. The instructions can be written in many different types of languages from high level languages to machine languages.

PROM.

An acronym which stands for Programmable Read Only Memory. These chips are programmed after they are manufactured rather than at the time of manufacture as with ROM chips.

proofreading.

The task preformed in the publishing process just before the mass printing. Proofreaders examine the text carefully to find errors in spelling, grammar, and punctuation, as well as other details which make the final copy correct and easy to read.

puck.

A part of the graphics tablet input device. A graphics tablet can employ either a stylus or a puck for data input. The puck looks like a mouse and is moved over the tablet surface usually to control the cursor on the display device screen. The puck usually has a button or buttons on its top side and felt on its bottom.

RAM.

An acronym standing for Random Access Memory. This is the working memory of the computer; memory where user generated programs, application programs, and data are stored. RAM memory is volatile, that is, data and programs stored here are lost when the computer is turned off or when power is interrupted.

raster refresh.

The most common type of display device used in computer graphics. A cathode ray tube (CRT) device whose electronic beam excites the phosphors on the screen by scanning the screen in a fixed pattern--each row from left to right and from top to bottom. Picture elements (pixels) are lighted according to a corresponding memory location in the computer's frame buffer. This technology is very similar to commercial T.V.

real time animation.

Used to describe animation sequences that are presented by the computer at a speed that makes their movement appear as smooth and fluid as it would be in real life. The normal alternative would be to have individual frames transferred to motion picture film or another media and then presented at a rate greater than 18 to 24 frames per second. This term is sometimes used to refer to the animation that occurs with interactive applications such as flight simulation where new scenes must be generated according to the reaction of the user.

relative coordinates.

A set of Cartesian coordinates that are given in relation to the last set of coordinates addressed rather than in relation to the origin as with a set of absolute coordinates.

rendering.

An artist's representation of an image or the act of creating such a representation.

resolution.

The realism, fineness, and smoothness which can be achieved when an image is represented on a display device screen or another output device. With a raster type device, resolution is dependent on the density of or size of the pixels. The more dense, the smaller the pixels, or the more pixels or points per inch, the better the resolution.

ROM.

An acronym standing for Read Only Memory. A solid state chip on which data or programs are stored at the time of manufacture and cannot be changed by the user. These chips are non-volatile, that is, they do not loose the data stored on them when the power is turned off.

RUN.

An Apple command (also used in many other operating systems) that causes the program currently stored in RAM to be executed. If used with a line number following, the program begins executing with that line number. If used with the name of a program which is not presently in RAM, the program will be loaded from the diskette in the disk drive and then executed. An error will occur if DOS or ProDOS has not been loaded into the computer memory.

SAVE.

An Apple DOS or ProDos command that saves the program currently in RAM to the diskette in the designated disk drive. The name under which the program is saved must follow the command. The last disk drive used will be used unless the user specifies another. An error will occur if the user attempts to use this command when DOS or ProDOS has not been loaded into the computer's memory.

scanners.

A computer image processing input device. A scanner is used to transfer photos, sketches, maps, drawings, and text into digital data that can be used in drawing, painting, and desktop publishing application programs.

shaded images.

Two dimensional depictions of 3D objects where variations in the amount of light striking the object are emphasized to give the illusion of 3D.

simulation.

Simulation is the use of a computer to represent certain features of another physical or abstract system. One of the best known simulations performed by a computer is that of flight. Computer controlled flight simulators are used by the military and the major airlines for the training of pilots.

snap-to.

A type of input constraint used in graphics software such as CAD and desktop publishing applications. A location which is being input by the user is constrained to fall on a grid intersection or on a point that is part of an existing entity. An example of the latter would be when a location being input is constrained to fall on the center of an existing circle or the endpoint of an existing line. Applications usually allow the user to control the snap-to constraint.

software.

Sets of programs which are sets of instructions for the computer. The instructions that govern how the computer functions. Software can include application programs, system programs, and utilities.

solids modeling.

A method of designing objects that creates a data base which contains all the attributes necessary to describe the object as a 3D geometric solid. Other methods of modeling include wire frame and surface modeling.

source code.

Refers to a program in its original form before it has been processed by an assembler, compiler, or interpreter.

storage tube.

A type of computer graphics display device that was very popular before microcomputers were so prevalent. It used a CRT with very persistent phosphors or it saved the image as electrical charges --once an image was drawn it would remain on the screen indefinitely. Images were drawn using the directed beam method, that is, the beam followed the path of the vectors rather than traveling in a fixed path as with a raster refresh device. The storage tube had excellent resolution, was low cost, and compatible with time sharing systems.

storyboard.

A graphical outline of a film or computer generated animation sequence. A collection of single pictures, each of which represents a distinct sequence or narrative element within the animated sequence.

text fonts.

A term used in publishing to describe a complete set of alphabetic, numeric and special characters all of the same size and style (design). Examples: 12 pt. Helvetica, 14 pt. Times Roman, and 18 pt. Times Roman are all different fonts.

thaumatrope.

An early animation device invented by an English doctor, John Paris. Pronounced THAW-ma-trope, the device was a disk with strings attached with which to spin it and two pictures, one on each side. As the disk was spun between the strings, the two pictures appeared to be superimposed on each other.

thumbwheels.

A computer graphics input device used to control the screen cursor. Normally, there are two wheels, one to control the horizontal cursor movement and one to control the vertical cursor movement.

track ball.

A computer graphics input device used to control the screen cursor. Consists of a ball mounted in a housing. As the ball is rotated, the screen cursor moves at the speed and in the direction of the track ball. Technically, it is an upside down mouse.

transformation.

In computer graphics this term refers to changes made to the data base which describes the image being drawn. These changes are made mathematically. The results cause the image to be moved around on the screen, rotated, drawn larger or smaller. The basic transformations in computer graphics are scaling, rotation, translation, reflection, shearing, and perspective.

viewport.

A computer graphics term that refers to that portion of the display device screen that is being used to display the selected image (window).

window.

A computer graphics term used to describe that portion of the data base (image) that has been selected to be displayed within the display device's viewport.

wire frame model.

A computer graphics representation of a 3D image where lines are used to describe the edges of the solid figure or to form a wire mesh around a curved surface. The image appears in skeleton form, very often without hidden lines removed.

workstation.

A classification of computer between a microcomputer and a minicomputer. Usually refers to a 32 or 64 bit machine that is commonly used as an engineering workstation. Most workstations have enhanced graphics capabilities.

WYSIWYG.

An acronym that stands for What-You-See-Is-What-You-Get. A term used in desktop publishing to describe the capacity of the display device to closely display the same image that will be printed. Pronounced WHIZ-ee-wig.

X axis.

The horizontal base line in a Cartesian coordinate system. X coordinate distances are measured parallel and Y coordinate distances are measured perpendicular to this axis.

Y axis.

The vertical base line in a Cartesian coordinate system. Y coordinate distances are measured parallel and X coordinate distances are measured perpendicular to this axis.

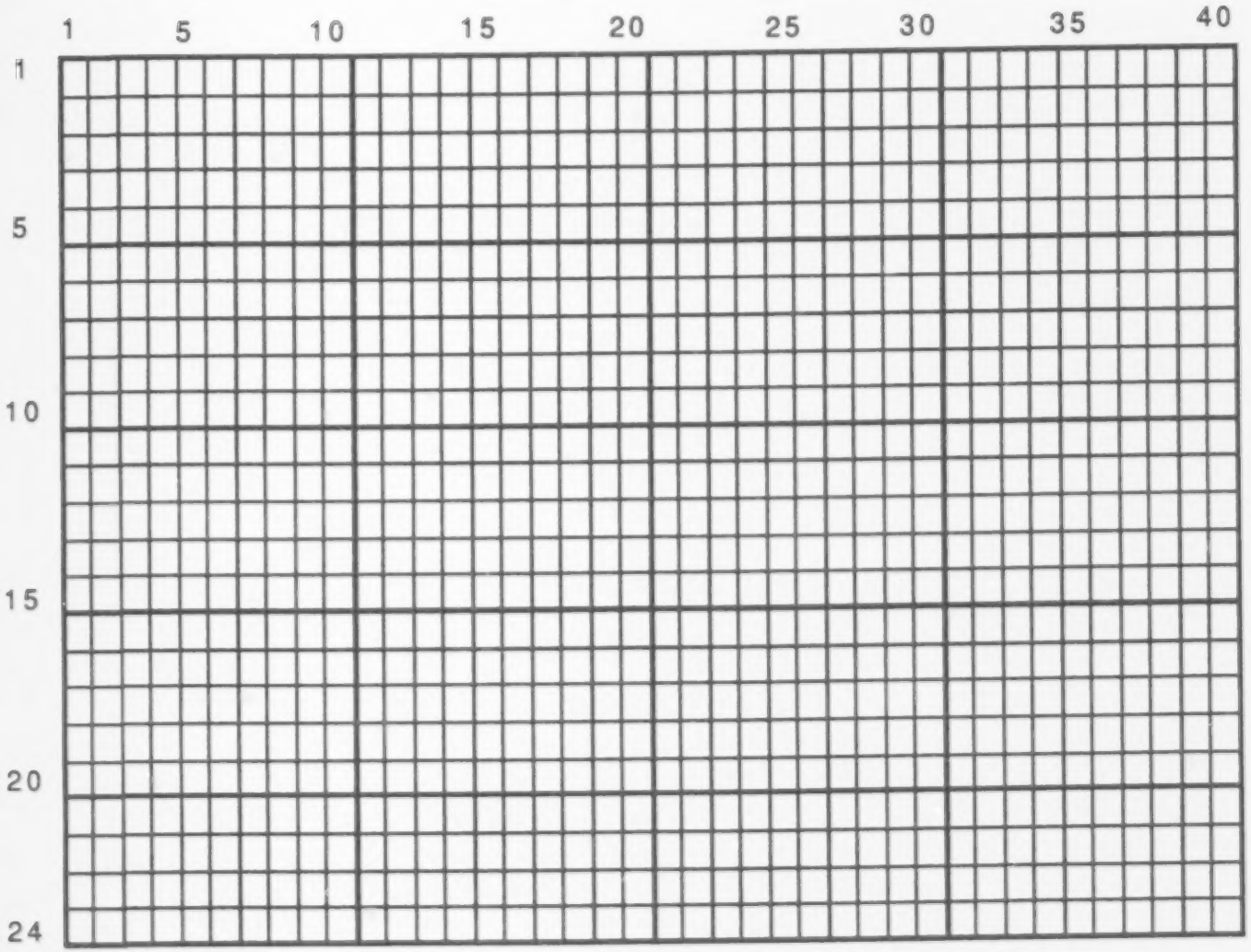
zoom.

The function in graphics application programs (CAD, Desktop Publishing, etc.) which moves the image closer or farther away from the user by increasing or decreasing the size of the window.

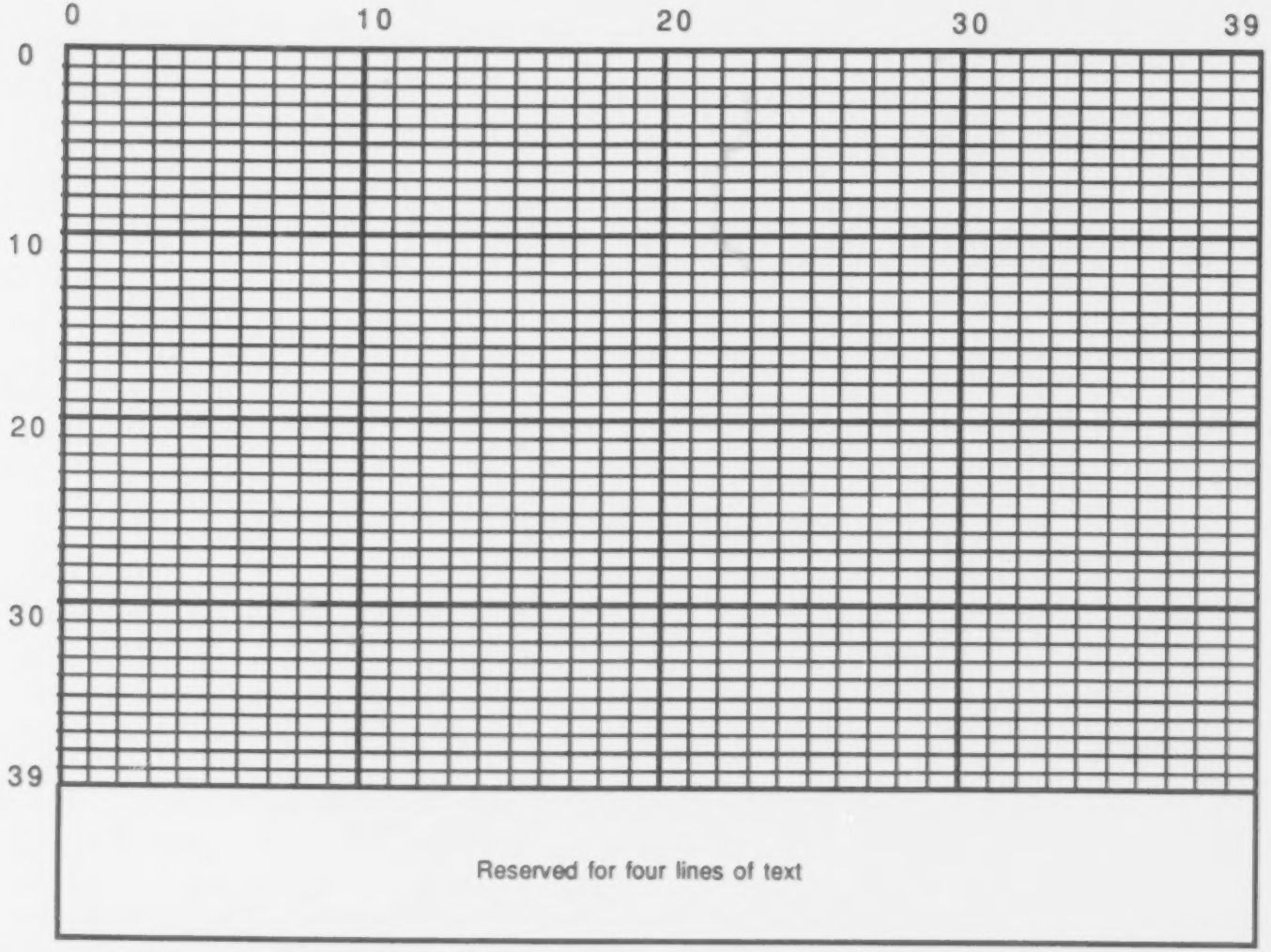
Appendix A

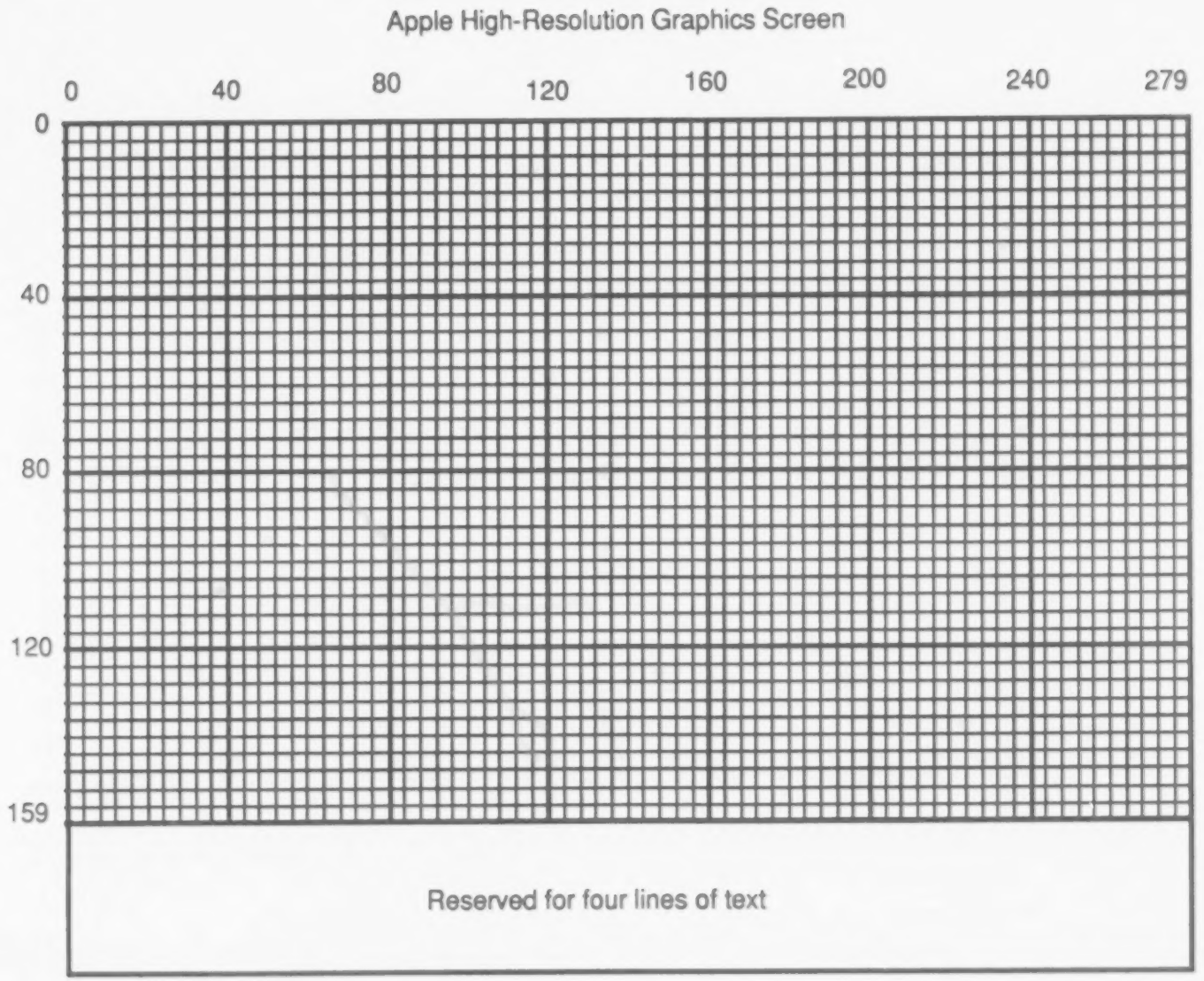
Apple Screen Grid Sheets

The Apple Text Screen



Apple Low-Resolution Graphics Screen




Apple High-Resolution Graphics Screen
0
40
80
120
160
200
240
279
0
40
80
120
159
Reserved for four lines of text

Appendix B

Equipment List

The following equipment is listed as an example of what could be purchased for a **new** laboratory. Individual schools will need to modify this list based on existing equipment and budgets. Before any purchases, teachers should check with their Apple dealer to learn about special educational discounts, new equipment that has been introduced since this list was compiled, and special hardware and software needed to connect and **network** this equipment.

Item	Number
Computer Workstation, Apple IIgs with 512k RAM, keyboard, mouse, Apple RGB monitor, 3.5 inch disk drive, and 5.25 inch disk drive	One per student + 1*
Printer, Apple Imagewriter II networked with 4 computers	One for every 4 computer workstations
Printer, Apple Imagewriter II	1*
Laser Printer, Apple Laserwriter II NT	1
Plotter, Hewlett Packard 7470A or 7475A	1
Tables, 3' x 4' computer	One per computer + 1 per printer + 2

* computer workstation and printer designated for the teacher.

If resources permit, consideration should be given to adding additional 3.5" disk drives and memory to the computers, modems, a scanner, and a large screen computer projection system to the above list.

Computer graphics laboratories can be set up using other Apple computers or other brands of computers. However, if this curriculum guide is to be used without modifications, Apple computers are the best selection and if new computers are being purchased the Apple IIgs is recommended.

Appendix C

Supply List

The following list of supplies is suggested for those laboratories where at least one of each type of equipment suggested in Appendix B is being used. Quantities ordered will depend on the size of the laboratory and number of classes being taught.

Item

Paper, Printer

Pen fed, 9.5" x 11" fan fold

8.5" x 11" copy paper, 16- to 24- pound bond

Paper, Plotter

"A" size (8.5" x 11"), 20 pound opaque white bond

"A" size, vellum

"A" size, polyester film

"B" size, 11" x 17", same as above (if lab is equipped with "B" size plotter).

Diskettes, Floppy

3.5" microdiskettes, double sided

5.25" single sided, double density

Pens, Plotter

Disposable fiber-tip for paper, black

Disposable fiber-tip for paper, assorted colors

Disposable fiber-tip for glossy film, black

Ribbons, Imagewriter II

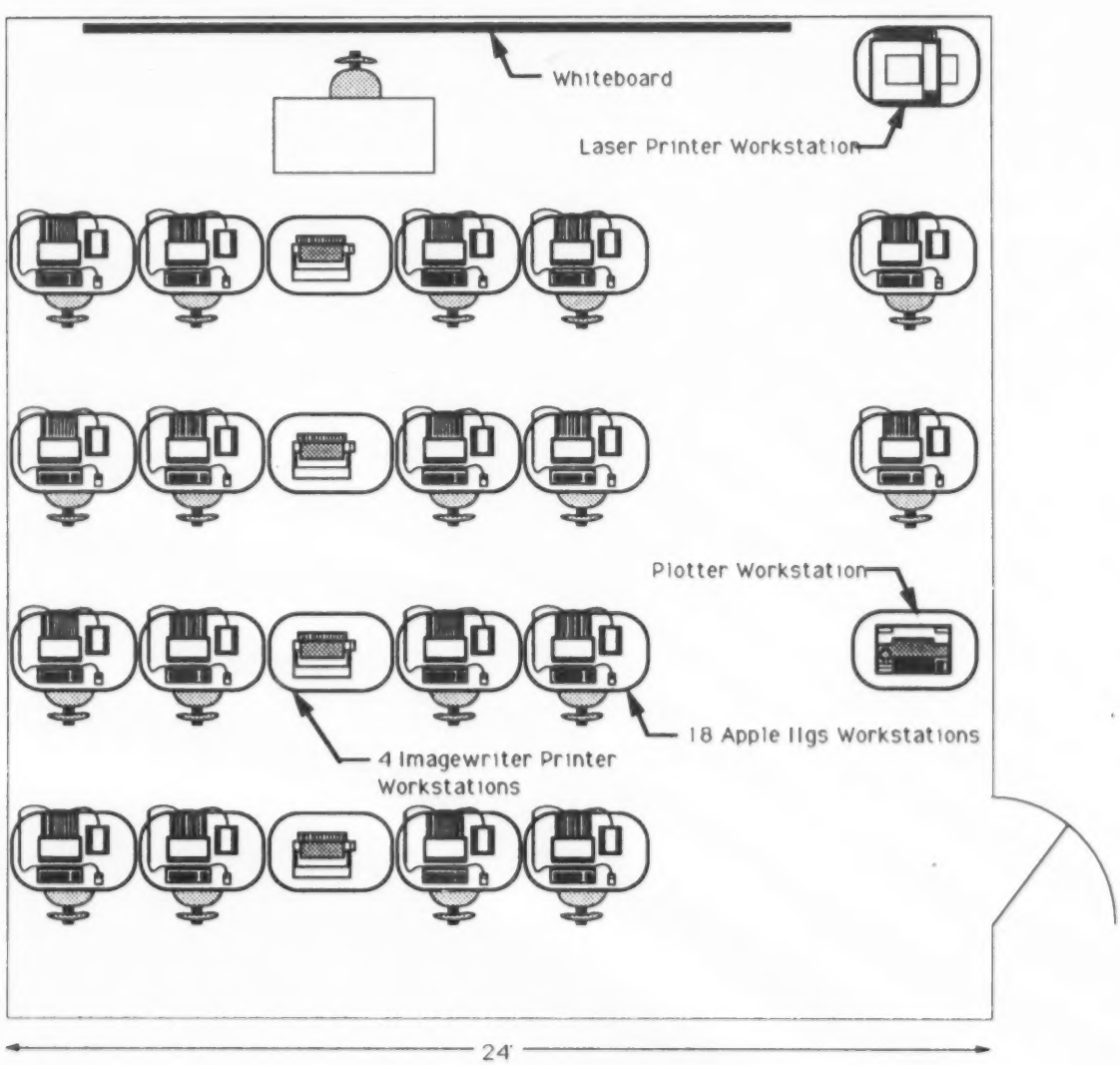
Continous loop fabric ribbons, black

Continous loop fabric ribbons, color

Toner, Laserprinter

Appendix D

Facility Layout



Note: This is just one possible solution for a computer graphics facility layout. For additional solutions and for more details concerning this layout, consult your computer vendor and the technology education supervisor in the Kentucky Department of Education.

To help improve the quality of this curriculum guide. . . .

. . . your comments would be appreciated, should you have suggestions about how the guide can be improved or if you find a mistake. When you make a suggestion, it would also be helpful to know how you are using the guide. Therefore, please supply the following information with your comments:

Name_____School _____

City _____ State _____

Telephone ( ) _____-_____

Name of course in which guide is being used _____

Grade level/s _____ Brand of computer _____

Model: II, II+, IIe, IIc, IIgs, other _____ Computer's RAM memory _____K

Type of printer _____ Number of disk drives per workstation _____

Type of plotter _____ Type of mouse _____

Software _____ _____

_____.

Comments (include page numbers if applicable):

Send comments to:

Clyde O. Craft
Department of Industrial Education and Technology
Eastern Kentucky University
Richmond, KY 40475-3115

For borrowing videotapes. . .

If your budget will not allow you to purchase the videotapes suggested in this guide, it is possible to borrow them from the author on a limited basis. If you wish to borrow one for a short period of time, supply the following information with your request and send it to the address below:

Name _____ School _____

Address _____ City _____ State _____

Videotape title _____

Dates requested for viewing (please allow plenty of lead time and give at least three possible dates):

_____ , _____.

I agree to return the videotape/s as quickly as possible and to not copy it/them.

_____
(signed)

Send to:

Clyde O. Craft
Department of Industrial Education and Technology
Eastern Kentucky University
Richmond, KY 40475-3115

For requesting the *Computer Graphics Curriculum Project Master Diskettes* . . .

The *Computer Graphics Curriculum Project Master Diskettes* contain files of those programs that students are to be given for certain exercises as well as files of each of the programs students are expected to enter into their computer in COG.003 through COG.010. The latter set of files is included for the purpose of helping the instructor who does not have time to work through the curriculum guide before making a presentation to his or her students. They are not intended for use by students. Therefore, caution should be taken not to give them to students as this would detract from the educational value of the activities.

Please send me a copy of the:

☐ DOS 3.3 version of the *Computer Graphics Curriculum Project Master Diskettes.*

☐ ProDOS version of the *Computer Graphics Curriculum Project Master Diskettes.*

Name _____ School _____

Address _____ City _____ State _____

Send this request to:

Curriculum Development Unit
Office of Vocational Education
2024 Capital Plaza Tower
Frankfort, KY 40601